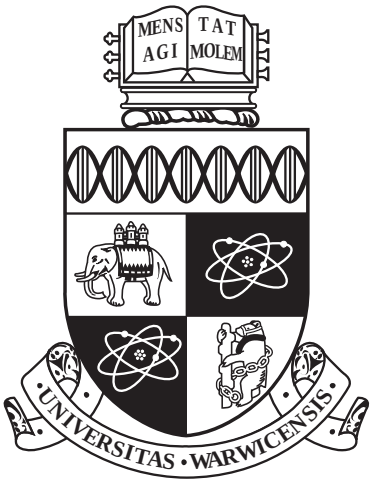


Polyomino Models Of Molecular Monolayers

by

Joel Nicholls

Thesis

Submitted to the University of Warwick

for the degree of

Doctor of Philosophy

Centre for Complexity Science

August 2017

THE UNIVERSITY OF
WARWICK

Contents

List of Tables

List of Figures

Acknowledgments

I would like to thank all the people involved in the Centre for Complexity Science who through our shared experience and many stimulating exchanges have shaped the past few years of my life and, undoubtedly, the research that has gone in to this thesis. Names I feel compelled to mention are Federico Botta, Diana Khoromskaia, Tom Machon, Jonathan Skipp, and Carl Whitfield, for the frequency of ideas exchanged, although the full number of supportive characters is much greater. I am very grateful and indebted to my two supervisors David Quigley and Gareth P. Alexander whose accurate advice, enduring support, and many inspiring discussions have been integral to the research undertaken. Lastly, but not in the least bit least, I must thank Stephen, Sue, Chloe, and Kanako.

Declarations

This thesis is submitted to the University of Warwick in support of my application for the degree of Doctor of Philosophy. It has been composed by myself and has not been submitted in any previous application for any degree. Part of the material in Chapter 5 has been submitted for publication and is currently under review:

- Joel Nicholls, Gareth P. Alexander, and David Quigley, Polyomino Models of Surface Supramolecular Assembly: Design Constraints and Structural Selectivity, arXiv:1702.01994 [cond-mat.soft]

Abstract

In this thesis, we describe periodic 2D supramolecular networks using a simple polyomino model with nearest-neighbour interactions. In particular, we focus on design rules for ordered molecular tilings, describing how the realised molecular tiling depends on the parameters of the system, such as the interactions, molecular shape, temperature, and defects. A major component of our analysis is in keeping the interaction parameters free and exploring the polyomino system from the perspective of the space of interaction counts.

The design principles and methods outlined in this thesis include several different themes, which give a complementary view on the properties of 2D supramolecular networks. Within the thesis we describe our algorithm for enumerating polyomino patterns and identifying their symmetries, making use of group theory methods that are specific for polyomino tilings. This information is used to analyse properties such as chirality, lowest energy states, degeneracy, and heat capacity curves, as depending on system parameters.

The domino tilings are considered in terms of enumeration by periodicity, and in terms of the kinetically accessible subsets of configurations. The generalisation to polyominoes gives a wider sense in which many of the techniques can be used, showing similarities and some differences with the domino system. A probabilistic version of algorithm DLX is described and tested that allows us to gather sample statistics of domino configurations for larger unit cells. Finally, fixed defects are considered and their effect on the kinetically accessible domino subsets is elucidated.

Chapter 1

Introduction

Recent experiments have seen the use of several techniques to create large ordered regions of molecular monolayers [1–6]. Several properties of the system are important for allowing these ordered regions to appear. Annealing of the system can be used to allow molecules to rearrange [6]. The right kind of anisotropy and strength of molecule-substrate and molecule-molecule bonds are needed to allow periodic patterns [2, 7]. Taking Ref. [8] as an example, from which example images are shown in Figure 1.1, the charge distribution around bromine atoms produces directional bonds between molecules and gold provides a relatively unreactive substrate. There can also be a subtle interplay between different factors, such as thermodynamics [9] and kinetics [3]. Apart from ordered domains, it has been shown in experiment that it is possible to form molecular networks which correspond to theoretically random tilings [10]. It is important to understand the factors that determine whether large ordered states can appear, to be able to design future experiments that are able to create them. The resulting ordered molecular monolayer gives an extremely thin surface with properties determined by the regular arrangement of molecules.

The general process of synthesising these 2D molecular arrays is often referred to as 2D supramolecular self-assembly. An overview of a wide range of self-assembling systems is given in Ref. [11]. Other examples include monolayers at near-full packing density [12] and ordered assemblies of millimetre-size polymers at the perfluorodecalin-water interface [13], which are also possible via self-assembly. The necessary conditions for self-assembly of a given molecular network depend on several different factors; for example, the combination of both dynamics and hydrogen bonding is critical for an explanation of the assembly of 1,4-substituted benzenediamine nanostructures on gold substrate [9]. Cyclohexa-m-phenylene can form different molecular networks, depending on whether it is adsorbed onto cop-

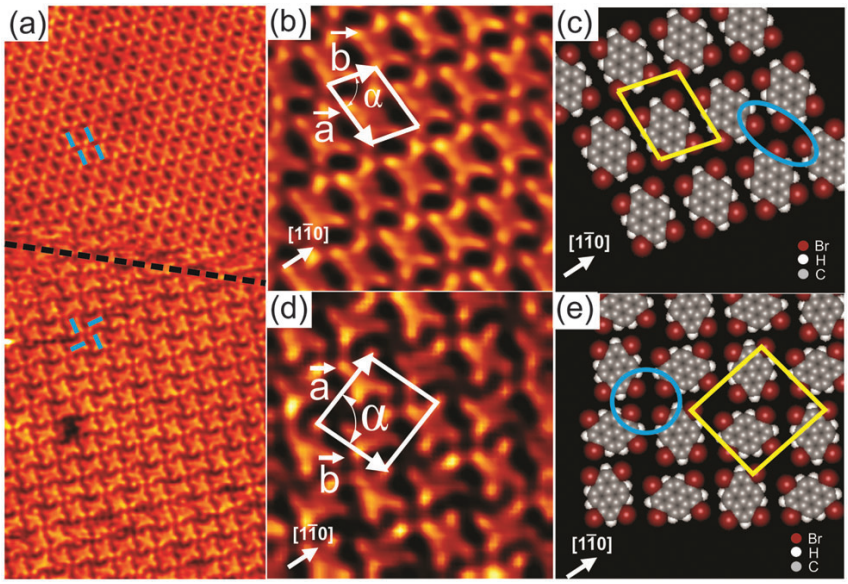


Figure 1.1: (a), (b), and (d) Scanning tunneling microscopy images for the pyrene derivative 1,3,6,8-tetrabromopyrene adsorbed to gold. Upper and lower images show two different patterns that can form. (c) and (e) give molecular models for (b) and (d), respectively. Reproduced from [8].

per, silver, or gold substrates [14]. The different kinds of network are shown in Figure 1.2.

A variety of techniques have been implemented to create and characterise self-assembled structures, such as deposition of molecules in ultra-high vacuum conditions, use of scanning tunelling microscopy to probe the structure, and the creation of regular molecular patterns via annealing [6, 8]. Another avenue of self-assembly research is in DNA origami, where 2D crystalline arrays of origami tiles have been built [15], and complex nano-scale shapes have been created out of self-assembled DNA strands [16].

Aside from space-filling tilings, several studies have explored rational design of shapes and structures from site-specific interactions, or by entropic self-ordering [17]. This includes complexity analysis of minimal sets of building blocks [18], the use of patchy particles to form quasicrystals [19], and self-assembly of charged soft dumbells [20]. Self-assembly principles are also critical for the production of many 3D structures, such as the viral capsids [21] illustrated in Figure 1.3 and supramolecular coordination complexes [22]. However, the design of specific interactions to assemble more complicated structures, say of lower symmetry, or with particular symmetries, remains a significant challenge.

Many kinds of metal-organic frameworks have been realised in monolay-

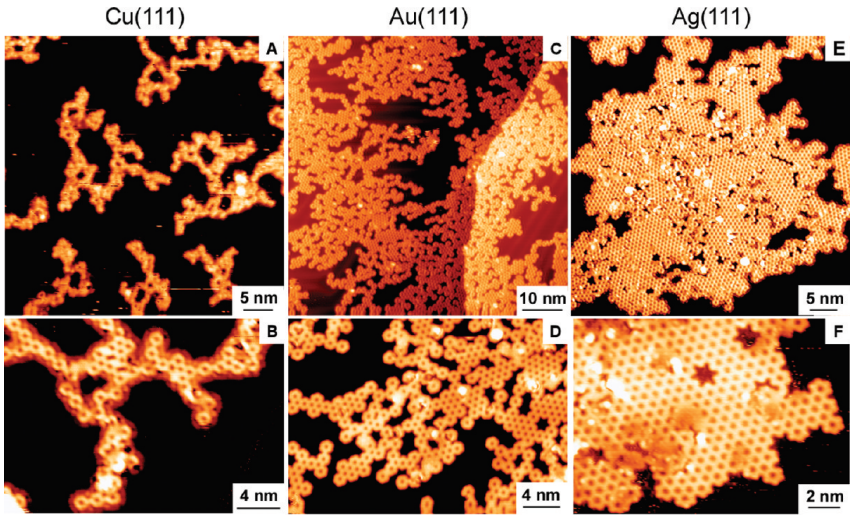


Figure 1.2: Scanning tunneling microscopy images of polyphenylene networks on copper, gold, and silver. Reproduced from [14].

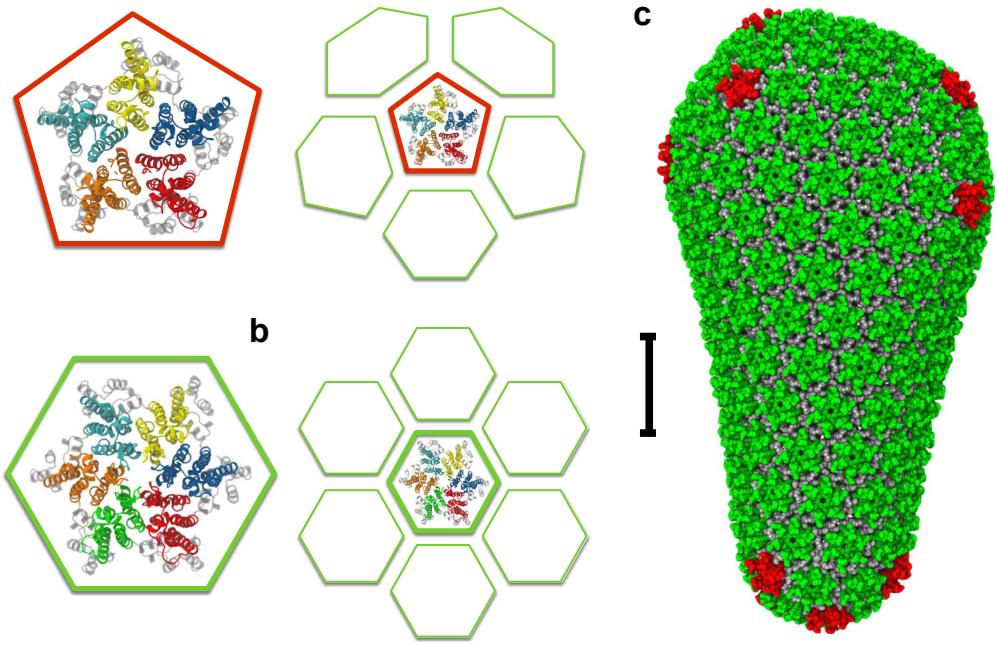


Figure 1.3: (c) HIV-1 capsid, assembled from (b) hexameric and pentameric capsid protein oligomer building blocks. Reproduced from [21].

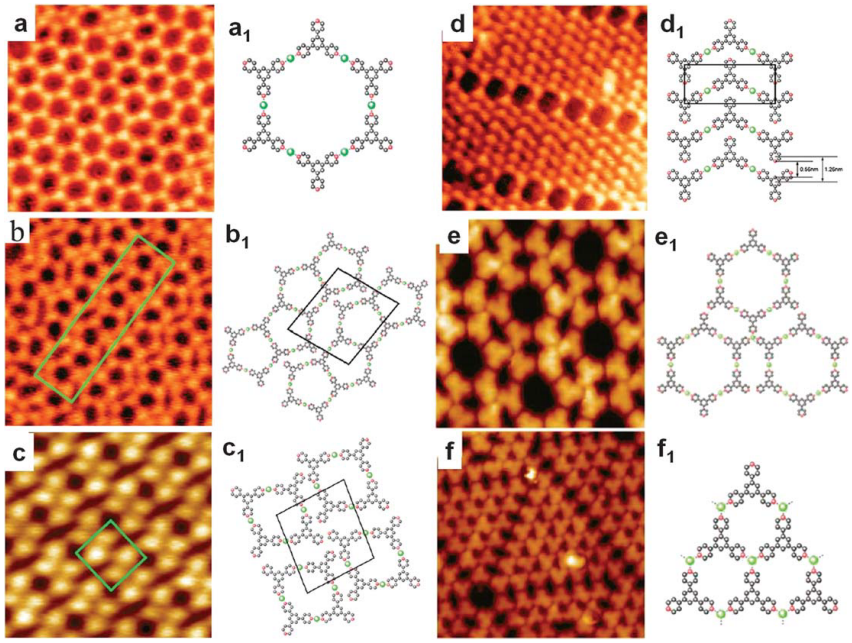


Figure 1.4: Various patterns of 1,3,5-trispyridylbenzene with Cu (TPyB-Cu) metal-organic frameworks on gold substrate, with density increasing alphabetically. Reproduced from [23, 25].

ers [23, 24]. These frameworks make use of molecules that are linked by metallic atoms, which gives rise to a plethora of different network structures, for example, the TPyB-Cu networks in Figure 1.4. In this case, the metal atom can be seen as a monomer with specific interactions. The experiment of Ref. [25] illustrates the principle that varying the experimental conditions will allow different molecular networks to be realised. For this system, molecular density dictates the type of pattern that appears.

Some of the most striking features of the self-assembled molecular monolayers seen in experiment are the symmetries of the patterns which are formed, for example, the cmm symmetry packing in Figure 1.5. The appearance of a particular translation or reflection symmetry in the thermodynamically favoured configuration does not generally have trivial dependence on the experimental parameters, for example it can be possible to have two coexisting molecular networks with different symmetries [8].

The substrate-molecule interaction strength is important for the self-assembly process. For example, molecular macrocycles can form monolayers on both graphite and gold substrates. On gold substrate, it was shown that a particular macrocycle can form an ordered network structure, whereas a similar macrocycle that bonds more strongly to the substrate was unable to form an ordered structure because

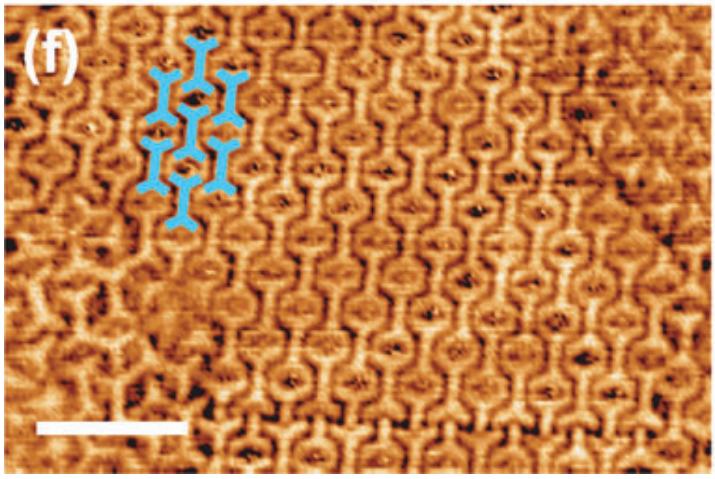


Figure 1.5: Dimerised 1,3,5-tris(4-bromophenyl)benzene (TBPB) on gold substrate, exhibiting a cmm wallpaper group. Reproduced from [12].

it was unable to undergo lateral reorganisation or adsorption-desorption [26]. The kinetic constraint in this case meant that the resulting configuration was not the most thermodynamically favourable. The more general problem is to find out the degree to which kinetic constraints affect the 2D molecular network.

When the substrate-molecule interactions are weak, the molecules can rearrange themselves more freely. Various heterocirculenes can form molecular networks on gold, but have weak interactions with the substrate [27]. For this reason, the molecules have few extra constraints and can form several highly symmetric patterns, such as the 4-fold symmetric pattern seen in Figure 1.6. However, in this case, the patterns are only achieved as separate islands and do not reach full monolayer coverage.

The recent exciting experiments involving the adsorption of a single layer of molecules onto a crystalline surface motivate our problem. It is often possible to test out different experimental conditions [4, 10] and observe that the molecules will assemble in different ways. However, it is more difficult to theoretically predict what kind of structure the molecules will assemble into, given specific experimental conditions. The resulting pattern is usually not known *a priori*. We would like to propose several methods to indicate the kind of structures that are likely to assemble, as well as other properties of the system of molecules. Furthermore, we suggest that specific structures could in principle be designed for, by adjusting the experimental set up. In particular, we are interested in the design of a particular pattern by selecting the interacting units that make up the monolayer. The fundamental rules

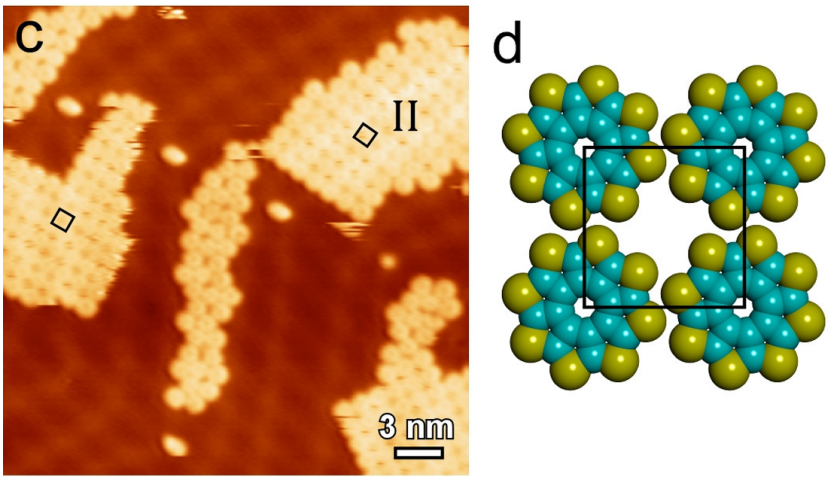


Figure 1.6: (c) Molecular islands of the Sulflower heterocirculene, displaying 4-fold symmetry, along with smaller disordered molecular aggregates. (d) Structural model of the Sulflower heterocirculene. Reproduced from [27].

that govern which pattern is produced by a single unit are called design principles.

It has been shown that changing the solvent, temperature, or molecular unit of a supramolecular network can alter the effective intermolecular interaction strength, leading to supramolecular networks with different structure, yet with molecular units of the same shape [10]. Furthermore, it has been recently demonstrated that large chiral domains can be formed from achiral molecules [2], as can be seen in Figure 1.7. Since many self-assembled systems have been shown to acquire useful or unusual properties, there is significant value in the rational design of self-assembled molecular networks.

Design principles have an advantage over the trial-and-error approach; for example, it can be predicted that 1,4,5,6-naphthalenetetracarboxylic diimide-melamine adsorbed molecules will not assemble a honeycomb array because of the periodicity of the underlying silver-silicon substrate [6]. This kind of reasoning can be used to inform the choice of new experiments.

Theoretical tools can be used to describe the relevant behaviour of self-asembled molecular systems. Molecular dynamics simulations are used for realistic time evolution of self-assembly [28], partition functions encode the thermodynamics of assembled systems [29], and investigation of assembly pathways provides new methods to generate ordered structures [30]. Furthermore, mathematical principles offer critical understanding, such as height functions to provide efficient computational sampling [31] and phase transition analysis to describe the temperature-dependence of system properties [32]. However, prediction of experimental results

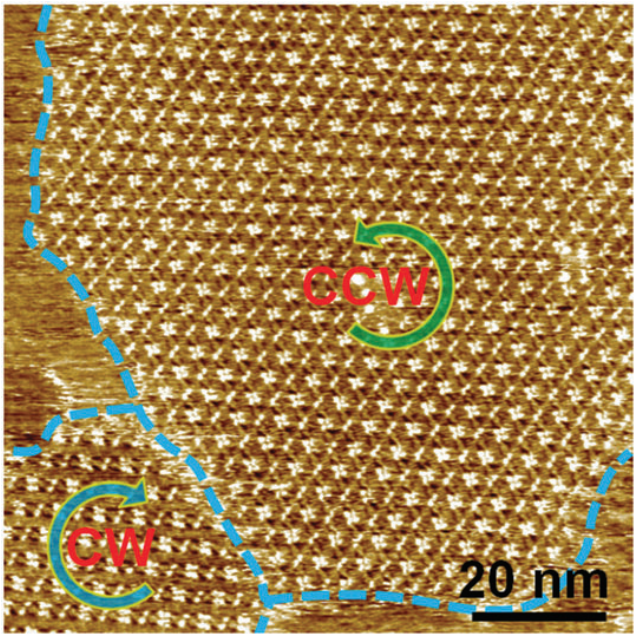


Figure 1.7: Monolayer of self-assembled 1,2,4-tri(tridecyloxy)-9,10-anthracenedione (TTA) molecules on highly oriented pyrolytic graphite (HOPG) substrate. Chiral domain boundaries are indicated by blue dotted lines. Reproduced from [2].

remains a challenge. Models that can be studied rigorously to predict structure and thermodynamics lack the chemical detail needed to connect to specific experiments, while on the other hand, the atomistic models that benefit from chemical detail are too computationally expensive to study rigorously.

In this thesis we have chosen to focus on a class of self-assembling 2D systems for which it is possible to enumerate low-periodicity configurations computationally. The parameters of the Hamiltonian can be chosen after computation, allowing us to explore the limits of the kinds of properties that can be designed for. Lattice models are well suited to this approach, where the chemical detail becomes coarse-grained. When molecular tilings are fully-packed, lattice models can be used effectively, since the molecules become interlocked. The domino system is perhaps the most well understood of all lattice packing problems [33]. In particular, the statistical mechanics of domino packings on the square lattice has been extensively studied since the work of Fisher [34] and Kasteleyn [35, 36], who gave an exact expression for the partition function of the purely entropic model (zero interaction energy) in terms of Pfaffians, and this work was expanded by others [37, 38]. The large system limit of the entropy per site is known, and provides the concept of interlocking entropy of dominoes. Subsequent studies have been primarily entropic or with achiral nearest

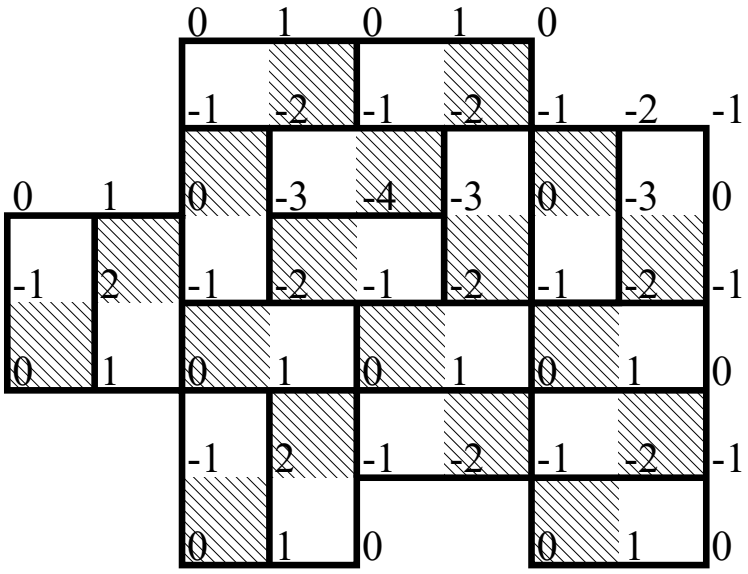


Figure 1.8: An illustration of a simply-connected, fully-packed region of dominoes, with corresponding height function overlaid. The hatcheted squares are even squares.

neighbour aligning interactions [39, 40].

For the fully-packed domino system, there is an equivalent representation as a height function [41, 42]. In the height representation, adjacent vertices of the underlying square lattice are given integer values such that moving anticlockwise around an even plaquette decreases the height by 3 when crossing a domino and increases the height by 1 otherwise. This gives a unique representation for each domino configuration as a set of vertex heights, apart from a constant shift to the heights at all vertices. An example is given in Figure 1.8 for a simply-connected region. Being simply connected means that any closed loop can be contracted to a point. For a periodic domino configuration, the change in height has a mean value along both vertical and horizontal directions, which we will refer to as the height change per plaquette (a 2D vector). As viewed from the height function representation, the height change per plaquette gives the criteria for domino configurations to be connected via flips, described in more detail in Section 3.6.

Not only have the static properties of dominoes been explored, but also the dynamic behaviour. Domino configurations on a simply-connected region are connected by series of domino flip moves [41], and these moves are the smallest possible rearrangements in a fully-packed system. Therefore, domino flip moves are a natural choice of dynamics. Furthermore, in the height function [43] representation of domino configurations, a domino flip changes the height only at the location of

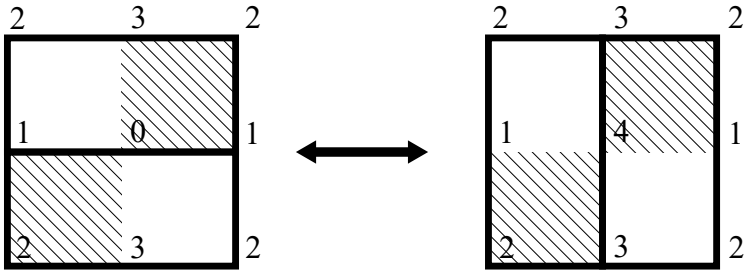


Figure 1.9: The domino flip move, with corresponding change to the height function. The value of the height function only changes at the centre of the domino pair being flipped.

the flip, shown in Figure 1.9. This in fact allows the domino configurations to be seen as elements of a distributive lattice [44]. Because of these properties, there are algorithms that generate random domino configurations quickly [45] and exactly [46]. In the algorithm of Ref. [46], coupling from the past is used to modify a Markov chain Monte Carlo method to produce exactly random samples.

Domino systems have yet more interesting properties. Exact values of domino pair correlations can be written down for the large system limit, by using perturbation of the Pfaffian corresponding to the partition function [47]. It is also shown that correlations between domino orientations in the asymptotic limit decays as a power law. In this sense, random tilings have short-range order due to full packing. Another consequence of the full packing of domino tilings is that the system has a strong dependence on boundary conditions. This can be seen most clearly for the Aztec diamond, displayed in Figure 1.10, which has a frozen region of dominoes outside the central circle [33, 42], constituting a nonzero fraction of the whole system. Interior boundaries are also an important feature of domino packings of simply-connected regions, which modify the connectivity under local moves [43, 48].

For molecular monolayers, there are often intermolecular bonds that promote ordered structures. In lattice model systems, this can be modelled by interactions between neighbouring sites. In the case of domino systems, interesting work has been done on attractive interactions between the long sides of two adjacent dominoes, which we will refer to as side-aligning interactions, shown in Figure 1.11. In this case, there is known to be a phase transition of Kosterlitz-Thouless type [49]. Interestingly, in each phase, the domino height function has the same average gradient of zero throughout the plane. But as explained in Ref. [50], the correlations in the low-temperature regime are markedly different than in the high-temperature regime, which can be seen in Figure 1.12. At high temperature, the dominoes are arranged mostly randomly, in a liquid phase. In this phase, the correlations between

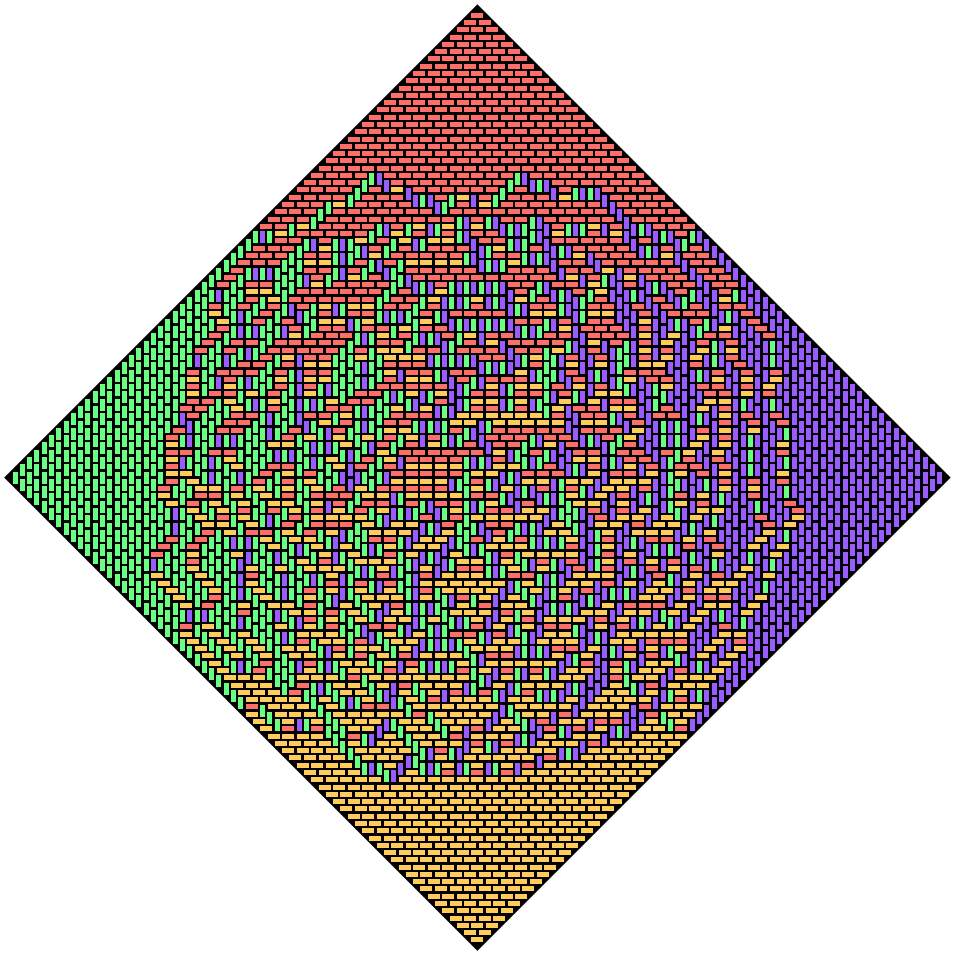


Figure 1.10: Random domino tiling of the Aztec diamond of order 64. The general form of the Aztec diamond boundary is four staircases glued together, with 2 squares at each corner. The colour of each domino is determined by the direction of increasing height function across the domino.

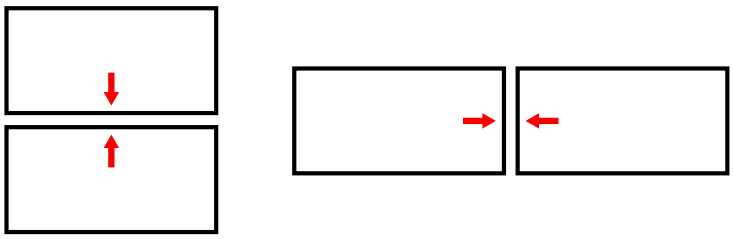


Figure 1.11: Two typical domino interactions, arrows signify attractive interactions between neighbouring dominoes. For both interaction types, configurations related by mirror symmetry are assigned equal energy.

dominoes fall off as a power law, with an exponent that varies continuously with temperature, so that dominoes are less strongly correlated at high temperature. For this system, a peak in the heat capacity indicates there is a phase transition. The heat capacity peak is located at a temperature close (but not equal) to the critical temperature, typical of a Kosterlitz-Thouless type transition [49].

The properties of molecular monolayers can depend strongly on whether the system is above or below a phase transition. For the side-aligning domino system below the critical temperature, the system enters a crystalline phase shown in Figure 1.13, where the domino-domino correlations extend indefinitely. This phenomena can be thought of in the context of the large scale behaviour of the height function of the domino tiling, which is affected by the interactions between dominoes. In the large scale limit, the height function approximates a continuous function, with a corresponding action to determine the behaviour at a given temperature [49]. At low temperature, a potential term promotes domino ordering. Conversely, at high temperature, an entropic term dominates, causing the height function to become rough. Indeed, in the high-temperature regime, the height fluctuations have the same Gaussian free field property as those of the non-interacting model [51]. This property in the interacting model has been suggested for some time, but was proved recently [52].

Interacting domino models apart from the usual side-aligning model have also been explored. Ref. [53] gives a version of the model where there are also attractive interactions between the short sides of dominoes, which we denote end-aligning interactions, illustrated in Figure 1.11. In this model, a nematic phase is possible only when a nonzero fraction of monomers are allowed. Anisotropic models have also been proposed [54], where dominoes have the usual side-aligning interaction, but have different interaction strengths for vertical and horizontal dominoes. Also,

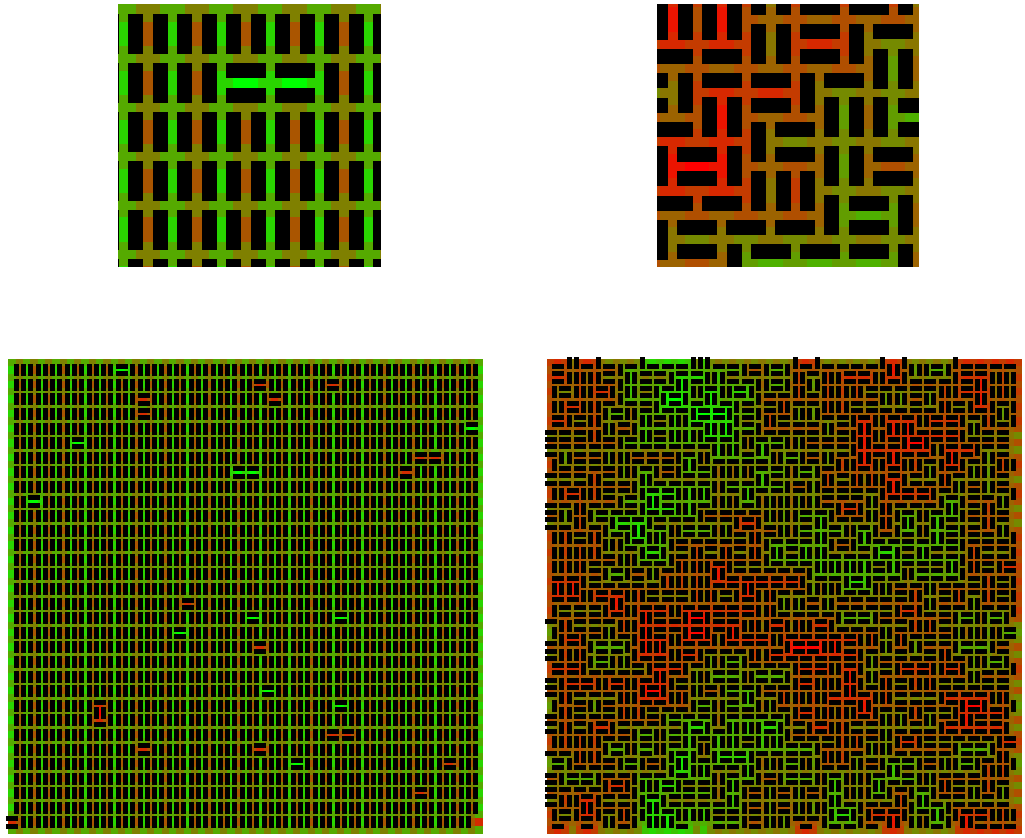


Figure 1.12: Side-aligning interacting domino configuration on periodic rectangular region of dimensions 64 by 64, at $\beta = 2.3$ (above the critical value), and $\beta = 0$. Dominoes are shown as black rectangles, and the background colour is given by the value of the height function at that point. Above each configuration is a zoomed-in picture. This simulation is performed for the set of domino configurations with zero height change per plaquette, equilibrated via domino flip moves.

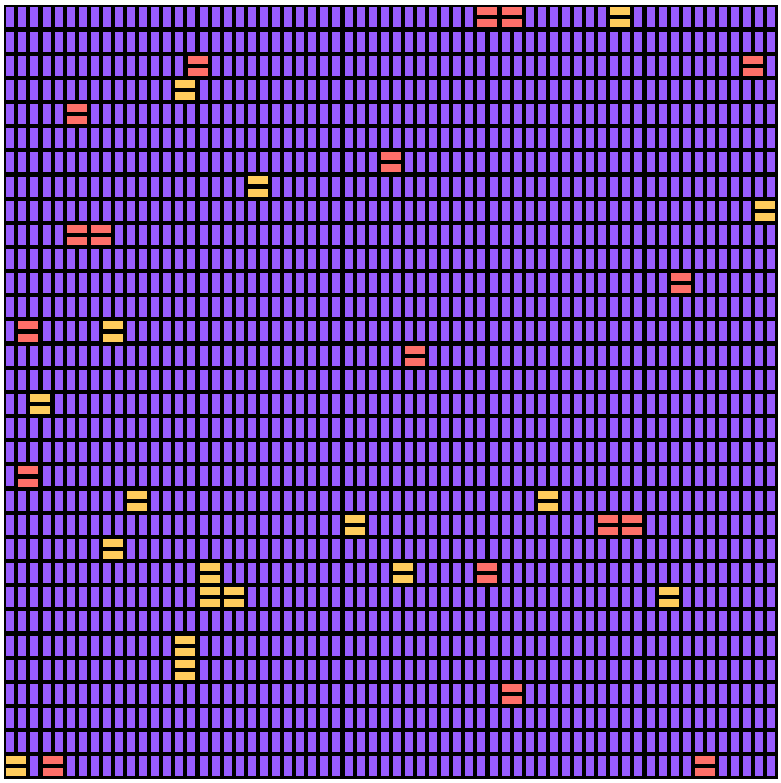


Figure 1.13: Side-aligning interacting domino configuration on a periodic rectangular region of dimensions 64 by 64, at $\beta = 2.3$. Dominoes are represented as coloured rectangles. Each domino is coloured according to the ground state it belongs to. There are 4 ground sates, which are degenerate as planar patterns. Since β is above the critical value, one colour dominates.

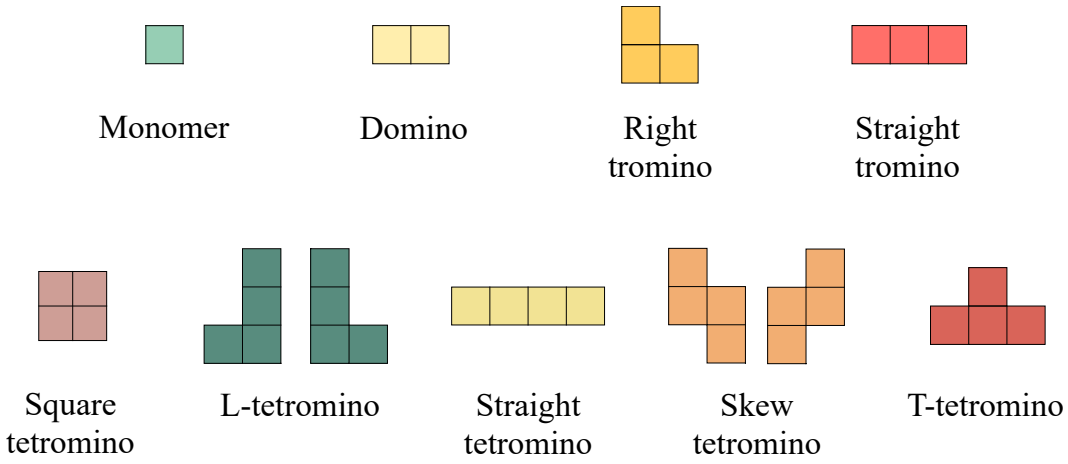


Figure 1.14: The 11 one-sided polyominoes with area up to 4. The polyominoes with area 3 are called trominoes and those with area 4 are tetrominoes.

the repulsive version of the side-aligning interaction is explored, which allows for a first-order phase transition [54]. In each of these cases, domino configurations that are related by mirror symmetry have equal energy.

It is natural to also consider the 3D analogue of the domino system. In this case, it is not possible to write down an analytic partition function. Therefore, some of the results from 2D domino systems are not applicable; for example, correlations between dominoes cannot be calculated by hand [55]. However, field variables can be defined, and the 3D random domino packings on the cubic lattice exhibit algebraic correlations [55]. Side-aligning interactions between dominoes have again been introduced, but in this case, a second order phase transition occurs, rather than a Kosterlitz-Thouless type transition [56].

Domino packing is greatly generalised by the problem of polyomino packing. A polyomino is a simply-connected set of squares of the square lattice, where each square must share at least one edge with another square of the polyomino [57]. Using polyomino shapes allows us to model more general molecules, which may pack differently than domino shapes. In this thesis, we discuss only polyominoes with area less than 5, which are shown in Figure 1.14. However, many of the techniques will also apply for larger polyomino shapes. We distinguish chiral polyomino pairs as being different. This means we are considering one-sided polyominoes [17] that are allowed to translate or rotate in the plane, but are not necessarily able to flip over. In contrast to previous studies of polyomino fluids [17, 58], we are most interested in the fully-packed configurations of a single polyomino type.

In random molecular networks, energetic interactions such as Van der Waals,

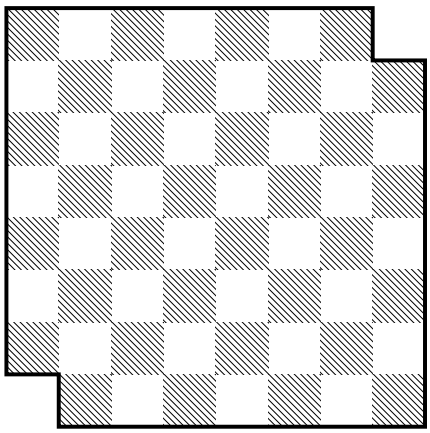


Figure 1.15: The chessboard with the squares from two opposite corners removed. This region cannot be tiled by dominoes, which can be understood by using a colouring argument [57].

hydrogen bonds, or halogen bonds are sufficiently weak that the entropy of configurations is important, and the molecules form networks that maximise their tiling entropy. The abstract mathematical problem of random polyomino tiling is well known in the literature and experimentally corresponds to the high-temperature limit of the systems we are considering. There are several techniques that can be used to analyse the fully-packed configurations, including: generating functions [59], transfer matrices [60], height functions [31], colouring arguments [61], tile invariants [62], local moves [41], Markov chain Monte Carlo Sampling [63], coupling from the past [46], and tile homotopy groups [64–66]. Some of these techniques have a long history, but have been further developed more recently. For example, an old puzzle involves the question of whether a chessboard with two opposite corner squares removed (as in Figure 1.15) can be tiled by dominoes. The answer is no, and can be understood using a colouring argument, making use of the fact that each domino takes up 1 light and 1 dark square. The overall number of light and dark squares in the region don't match, meaning there cannot be a tiling. This is the prototypical colouring argument, paving the way for more recent work on generalised arguments and more sophisticated tools of this kind [66].

Mathematical properties of random tilings depend on the type of polyomino being considered. For T-tetrominoes in particular, there are several known results. Firstly, one can quickly discover by enumeration that T-tetrominoes seem to only tile rectangles with both sides divisible by 4. The proof of this statement is non-trivial, but was completed in Ref. [67]. Furthermore, it was recently shown that for these tilings, a weighted partition function can be written down, which is related

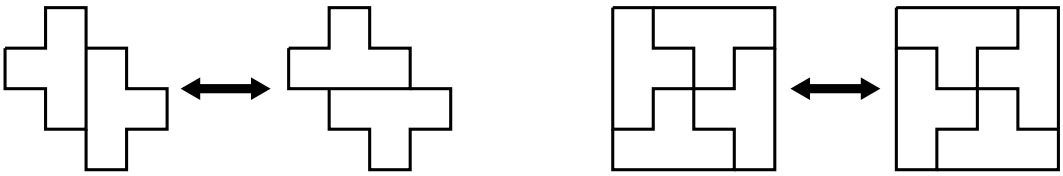


Figure 1.16: The two T-tetromino moves that allow any two configurations of a given simply-connected region to be connected by a sequence of moves. The original version of this figure can be found in Ref. [31].

to the Potts model [68]. Another important result is that there exists a set of 2 local moves that allow the entire set of tilings of a simply-connected region to be connected via a sequence of local moves [31], these moves are given in Figure 1.16. Connectivity under local moves gives an indication of how easily fully packed tilings can be rearranged.

There is also a natural choice of local moves for straight tromino configurations. In this case, moves involve 3 horizontal straight trominoes being replaced with vertical ones, as well as the converse operation. These moves are called flips, in analogy to the domino flip move. The straight tromino flip moves are ergodic on simply-connected domains and have an associated height function [60]. For fully-packed domino configurations, the height function gives all the information about whether configurations are connected by a sequence of domino flip moves. However, a similar result is not known for the straight trominoes. For right trominoes, there are known to be arbitrarily large simply-connected regions that can only be tiled by right trominoes in ways that differ everywhere on the entire region [62]. For this reason, it is generally not possible to make a sequence of small moves to rearrange one configuration into another. This gives a sense in which these tilings are quite inflexible to being rearranged via local moves.

Straight rectangular polyominoes of width 1 and length greater than 7 have several phases depending on density. In order of increasing packing density, there is a disordered phase, then a nematic phase, then finally a disordered phase as the system approaches full packing fraction [32]. This is one example that shows that behaviour at high (but not fully packed) density can be qualitatively quite different from the behaviour in the fully packed regime. Conversely, for right trominoes, the use of transfer matrix method shows there is no observed phase transition with density, and correlations are short-ranged [69].

Polyomino tiling has long been established as a computationally difficult problem. It is not easy to find short-cuts in the enumeration of polyomino patterns, except in certain special cases. Moreover, polyomino problems are computationally

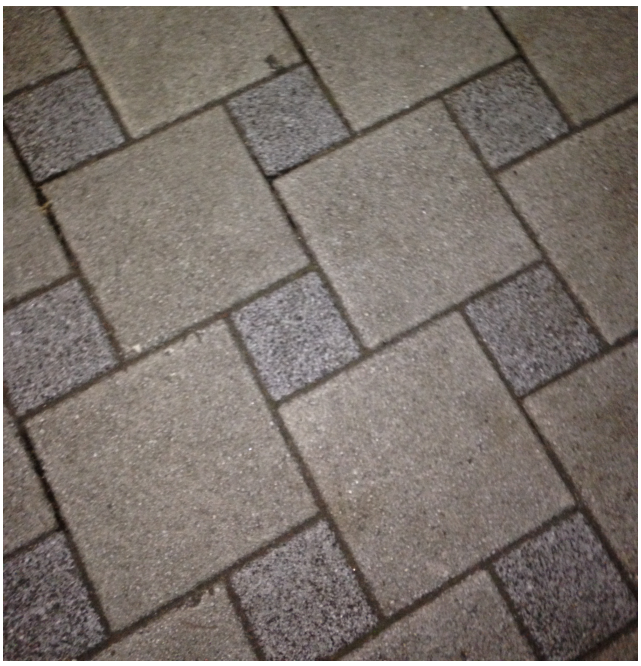


Figure 1.17: An example of a Pythagorean floor tiling. This tiling can be imagined as a mixture of square tetrominoes and monomers.

hard for several different reasons. The question of decidability for planar polyomino packing is a difficult problem in itself. Given a finite set of polyominoes, the problem of being able to tile the plane with them is generally undecidable [70]. Indeed, the problem is still undecidable when the set of polyominoes is restricted to 5 [71]. However, the problem of a single polyomino tiling the plane by translations is decidable [72, 73]. Another positive result is that a polynomial time algorithm exists for deciding whether a polyomino (along with its rotations and reflections) can produce an isohedral tiling of the plane [74]. An isohedral tiling is one in which any polyomino can be mapped to any other polyomino using the symmetries of the tiling. It is interesting to note that the decidability of tiling the plane with 1 polyomino (allowing for rotations and reflections) is still an open problem. Despite the theoretical difficulty of this problem, many different tilings are used in physical floor tilings with simple polyomino shapes, an example of which is shown in Figure 1.17.

Tiling a finite region with a set of tiles is NP-complete [75], and even more specifically, tiling with the set square tromino (with all 4 rotations included) is still an NP-complete problem [75]. For the problem of enumerating the set of all polyominoes of a given area [76, 77], the number of polyominoes diverges exponentially with the area of the polyomino, and there is no known analytic formula for this sequence. To obtain the entropy per area of random tilings, numerical estimates must

be used [59, 60], apart from special cases, such as the domino packings [35]. Importantly, the number of polyomino packings within a finite region typically grows exponentially with area [35, 59, 60], making enumeration of large regions computationally intractable.

In this thesis we will build on the concept of packed polyomino models to represent 2D molecular tilings. We add our own analysis of polyomino patterns and introduce new kinds of interactions. Previous polyomino models have mostly used a single polyomino type and Markov chain Monte Carlo methods or transfer matrices for sampling, whereas we use enumeration of low-periodicity polyomino patterns. We will generate polyomino patterns and identify their symmetries to explore principles for the formation of the various symmetric patterns found in experiments. The domino model is of particular interest, which previous authors have investigated with several kinds of interactions. We will explain design principles for a range of interactions, which expand on previous models by allowing for interactions that favour tilings of one handedness over the other. A large part of the existing literature on fully-packed dominoes makes use of a height function for determining correlations or for designing sampling algorithms. We will use the height function along with its topological definition [43] to identify subsets of domino patterns that are connected via molecular rearrangements.

Chapter 2

Theory

In this chapter, we introduce the interacting polyomino model that we use throughout the thesis to investigate the properties and design principles of 2D molecular networks. We also explain some of the established results on symmetries, wallpaper groups, and translational lattices that will be important for the discussion of polyomino symmetries later in the thesis. Finally, we introduce the concept of local moves for the rearrangement of polyomino patterns, in particular explaining the main ideas of the domino flip moves, relying on the work in Ref. [43] and adding an interpretation for planar domino patterns.

2.1 Interacting Polyominoes

Molecular networks in 2D are composed of a single layer of non-overlapping molecules. When they are packed together, each molecule creates a 2D region of excluded volume, which other molecules cannot overlap with. We model each molecule as a simple shape that approximates the true excluded volume. Indeed, experimental researchers often describe or depict molecular networks as being composed of simple shapes, representing the molecular units [2, 10].

We use polyominoes as our simple molecular shapes. To represent a high-density molecular network, we make use of fully-packed polyomino configurations on the square lattice with periodic boundary conditions to mimic planar tilings. The other natural choice of boundary condition is the rigid boundary, which polyominoes must not cross. However, using rigid boundary conditions would introduce edge effects [47, 78], as well as restrict the configurations to a subset belonging to only one height change per plaquette vector [43]. For other polyomino tilings, similar unwanted effects would occur, for example the skew tetromino does not tile any

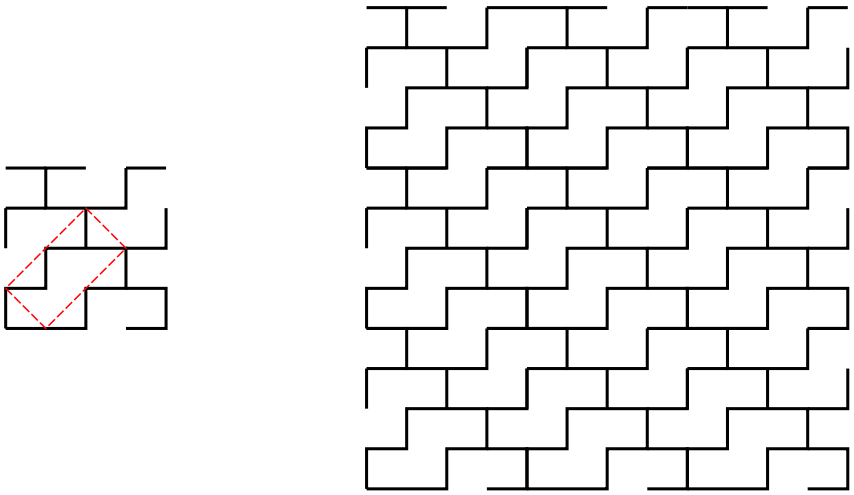


Figure 2.1: Example tiling of the plane by the skew tetromino. On the left is a square unit cell with periodic boundary that can be used to tile the pattern shown on the right. Note that here, the square unit cell is not of minimal area. A unit cell of minimal area (known as a primitive unit cell), is shown as a dashed red rectangle.

rectangle [79] at all (even allowing for the reflected shape), but it clearly can tile the plane, as in Figure 2.1.

Another reason to use periodic boundary conditions is that the lowest energy configurations of the plane will often be doubly-periodic, which can be seen in large yet finite experiments [6, 23, 80]. For 2D patterns, doubly-periodic means that the pattern is periodic in 2 linearly independent directions. Every doubly-periodic configuration of the plane can be represented by a configuration on a rectangular unit cell with periodic boundary that tiles the plane to recreate the aforementioned planar configuration. Therefore, we use periodic rectangular regions to enumerate configurations and find periodic lowest energy states. Furthermore, by enumerating regions of increasing size, it is possible to recreate all configurations of low periodicity - the details of this are set out in Subsection 3.4.2.

2.1.1 Interactions

Neighbouring molecules in the monolayer can attract or repel each other via weak, short-ranged interactions. For our model, we make use of effective interactions between faces of neighbouring polyominoes. These effective interactions represent weak, short-range bonds, such as Van der Waals [3], hydrogen bonds [4], or halogen bonds [8]. The interaction strength is determined by the type of bond, the actual distance between molecules, and the specific molecules themselves.

By using interactions between faces of neighbouring polyominoes in our model, the directionality and specificity of bonds can be taken into account. Formulating the bonds as interactions between faces in contact gives a general way to describe interactions between neighbouring polyominoes, as the information about the relative position and orientation of neighbouring polyominoes is encoded by the information on which faces are in contact. Furthermore, framing the problem in terms of interactions of faces in contact allows for simple computation of configurational energy.

In this thesis, we will consider only the intermolecular interactions. We assume that the interactions between molecule and substrate do not interfere with the molecular network that arises. This would mean the substrate-molecule interactions are not strongly varying with the position of the molecule on the substrate, or those interactions are weak compared to the intermolecular interactions, or the periodicity of the substrate and ideal molecular network happen to be commensurate. Intermolecular interactions alone can provide a rich variety of behaviour [50], and many of the experimental systems with interesting molecular networks can be approximated in this way [8, 10, 27].

2.2 Symmetry

Group theory is necessary for understanding the spatial symmetries of crystalline molecular networks and for analysing those symmetries. Furthermore, the topology techniques used in chapter 8 directly require group theory concepts. For these reasons, mathematical group theory is an important tool for the research in this thesis. Symmetric patterns can also be aesthetically pleasing, such as the tiling in Figure 2.2. There are many books on the subject of group theory [81], which give the central properties of groups. In this section, we outline some points about physical symmetry in two dimensions that will be important throughout the thesis.

2.2.1 Euclidean Symmetries

As mentioned, many experiments on 2D molecular networks exhibit ordered patterns consisting of many molecules, with translational symmetries in both directions. This state is crystalline because of its translational symmetry. Therefore, ordered molecular tilings have crystalline structure and are invariant under a group of symmetry operations.

To be clear, the full set of operations we would like to consider is the Euclidean group in 2D. These operations are made up of combinations of translations,

Figure 2.2: Roman mosaic with wallpaper group pg, in the Baths of Caracalla, Rome.

rotations, and reflections that correspond to the usual notion of spatial symmetry. The symmetries that preserve a given doubly-periodic, planar polyomino configuration form a discrete subgroup of the Euclidean group.

Each element of the Euclidean group in 2D acts on all points (x, y) in the plane as a bijective transformation, and can be written uniquely in the form

$$\begin{pmatrix} x \\ y \end{pmatrix} \rightarrow \begin{pmatrix} \cos(\theta) & -\sin(\theta) \\ \sin(\theta) & \cos(\theta) \end{pmatrix} \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}^n \begin{pmatrix} x \\ y \end{pmatrix} + \begin{pmatrix} a \\ b \end{pmatrix}, \tag{2.1}$$

where a,b are real numbers of any value, n is either 0 or 1, and θ is from 0 to 2π. The first matrix here is the rotation matrix and the second is used to introduce reflection. The symmetry operations are typically subdivided into 4 types. These are: rotations, translations, reflections and glide reflections. A visual explanation is given in Figure 2.3. The formula for each are:

- Translation: $n = 0$, $\theta = 0$
- Rotation: $n = 0$, $\theta \neq 0$
- Reflection: $n = 1$, $a\sin(\theta) = b(\cos(\theta) - 1)$
- Glide reflection: $n = 1$, $a\sin(\theta) \neq b(\cos(\theta) - 1)$

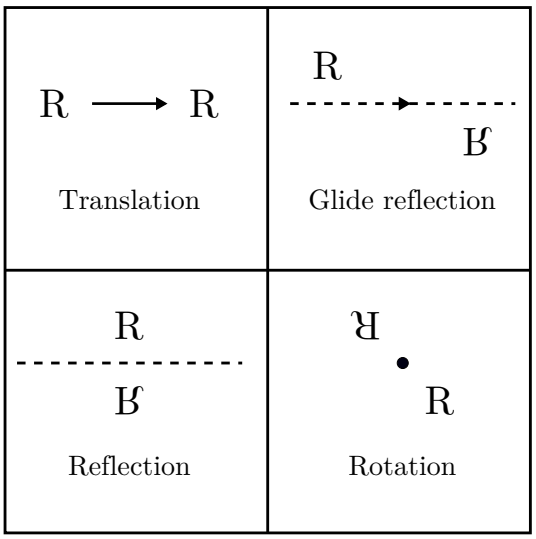


Figure 2.3: Visual examples of the 4 types of 2D Euclidean symmetries. The letter R is used to demonstrate how objects are affected by the transformations.

The difference between the reflection and glide reflection is that for reflection, the translational vector (a, b) must be perpendicular to the reflection axis; under this condition, the reflection axis is simply shifted. For the glide reflection, there is a translational component in the direction of the reflection axis. Therefore, glide reflection leaves no point of the plane unmoved.

2.2.2 Symmetries On The Unit Cell

The symmetry group of a doubly-periodic polyomino pattern on the plane is defined as the largest subgroup of the Euclidean group that preserves the polyomino pattern. However, we generate configurations on a rectangular unit cell (i.e. rectangular region with periodic boundary). Therefore, we must make use of a workable concept of symmetry on the unit cell.

Tiling the unit cell across the plane gives a planar polyomino pattern. This planar polyomino pattern has a symmetry group, as mentioned. For our configurations on rectangular unit cells, we will use the symmetry group of the corresponding planar pattern to describe the symmetry of our pattern on the unit cell. More specifically, the action of the Euclidean group on the plane descends to an action on the torus under the covering projection from the plane to the torus $R^2 \to R^2/(N\mathbb{Z} + M\mathbb{Z})$. The symmetry group of the unit cell is the image of the symmetry group of the plane tiling under this projection.

Suppose the edges of the unit cell are translation vectors $(N, 0)$ and $(0, M)$.

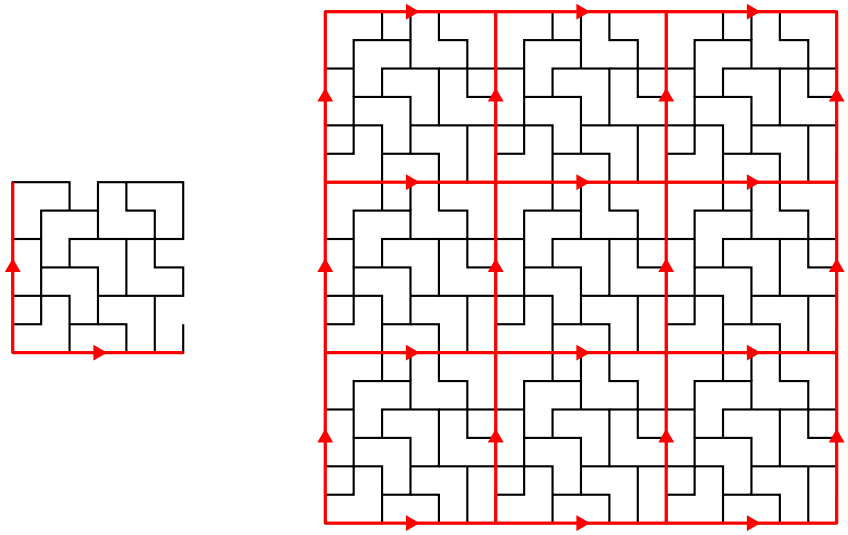


Figure 2.4: 9×9 Unit cell containing right tromino tiling, where red arrows signify unit cell translations. On the right, the unit cell is tiled, creating a pattern that is guaranteed the unit cell translational symmetries.

These two vectors define generators of a lattice of translational symmetries that the planar version of our polyomino pattern is guaranteed to have. An example is shown in Figure 2.4.

In practice, applying a symmetry operation to our unit cell can be done by first using the formula as used in Equation (2.1), then secondly by applying $\mathrm{mod}(N)$ to the x component and $\mathrm{mod}(M)$ to the y component. In this way, the image is wrapped around the periodic boundary at any location that it would leave the unit cell. This is illustrated in Figure 2.5.

When a pattern on the plane is left unchanged by a Euclidean symmetry, the corresponding pattern on the unit cell is also left unchanged. Therefore, we have a means for understanding symmetry for configurations on the unit cell, by comparison with the symmetry of the corresponding planar configuration.

The patterns on the $N \times M$ unit cell must always correspond to a planar pattern that has both translations $(N, 0)$ and $(0, M)$. In this sense, the dimensions of the unit cell place a lower limit on the symmetries of any planar pattern generated by that unit cell. By considering Euclidean symmetries with $0 \leq a < N$ and $0 \leq b < M$, we will not miss out on any symmetries except those that are equivalent to a symmetry operation that has the permitted values of a and b. Therefore, the restricted set of symmetries (along with the unit cell dimensions) provide all the information that could possibly be asked about the symmetry of the pattern.

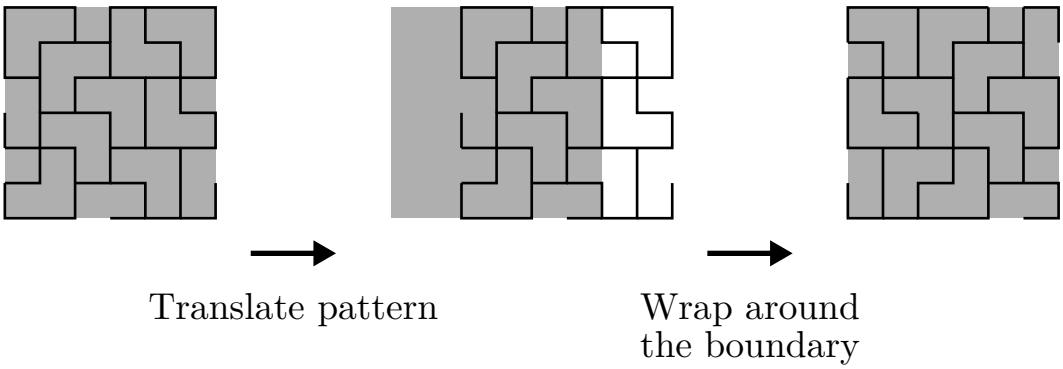


Figure 2.5: Translation applied to 6×6 unit cell of right trominoes. First, the transform is applied to the pattern, then the pattern is wrapped around the boundary by using $\mod 9$ operation on both x and y components of the part of the image that has left the unit cell.

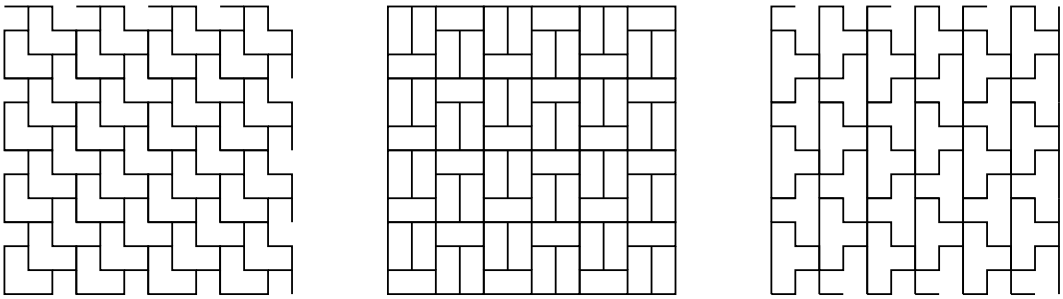


Figure 2.6: Examples of wallpaper groups corresponding to 3 polyomino patterns.

2.2.3 Wallpaper Groups

The translation, rotation, and reflection symmetries of a periodic tiling of polyominoes give us important information about that tiling, as mentioned in the introduction. One especially natural way to classify those symmetries is by assigning each tiling to a wallpaper group. The wallpaper group of a 2D pattern is a classification of the pattern by its symmetries and the way those symmetries are combined. Moreover, every doubly-periodic pattern can be assigned a wallpaper group uniquely. See Figure 2.6 for some examples.

To classify a pattern by its wallpaper group, it is sufficient to calculate the symmetry of the pattern as an abstract group. All patterns with isomorphic abstract symmetry groups will be placed in the same category of wallpaper group [82]. When all possible doubly-periodic patterns are considered, there are 17 such distinct abstract symmetry groups, shown in Figure 2.7.

Symmetry groups can also be considered to be equivalent as wallpaper groups

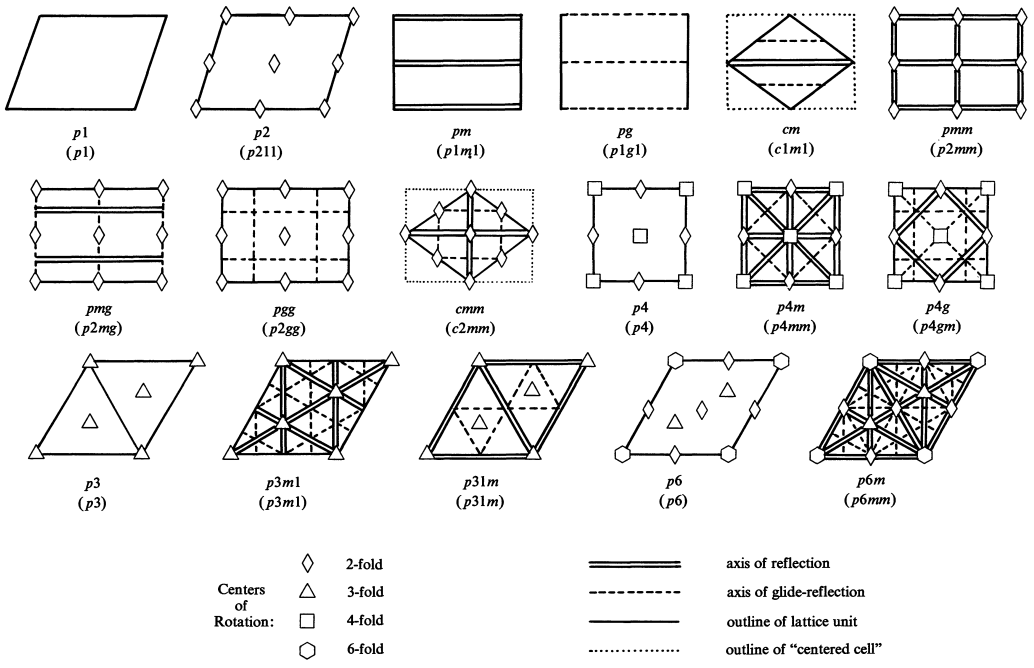


Figure 2.7: The 17 Wallpaper groups, described by the symmetry operations within a primitive cell. Reproduced from [83].

when they are conjugate under an element of the affine group (of which the Euclidean group is a subgroup). This seemingly stronger condition is actually equivalent in the case of wallpaper groups [82], meaning that the wallpaper group classification is a very natural way to classify the symmetry of 2D patterns. Figure 2.8 gives an idea of how polyomino configurations can be categorised as one of the 17 wallpaper groups that were shown in Figure 2.7.

2.2.4 Chiral Symmetry

Another natural way to classify periodic polyomino patterns is by the chirality of the polyomino pattern. As noted in Ref. [84] and in agreement with Ref. [85], a pattern is enantiomorphic if the motif cannot be superimposed on its mirror image by a 2D translation or rotation. This is the case when there are only orientation-preserving operations. Making use of this definition for our polyomino patterns, we use the term chiral for a pattern if its symmetry group contains only orientation-preserving operations, and denote the pattern as achiral otherwise. Note that if a pattern does not contain any mirror symmetries, it may still be achiral due to glide reflections. The pgg T-tetromino pattern shown in Figure 2.9 gives an example of a pattern that is achiral due to glide reflections; there is a reflection that is equivalent to a

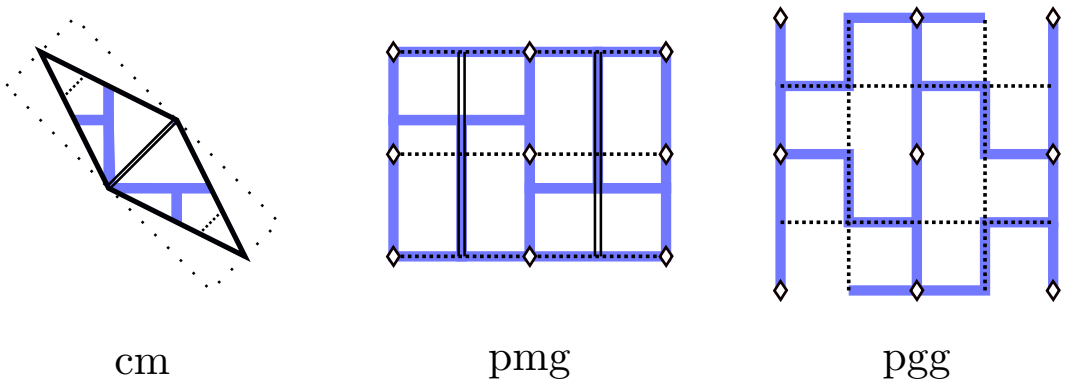


Figure 2.8: Examples of 3 polyomino configurations within their primitive unit cells. The polyomino pattern is shown by light purple lines. Dashed lines correspond to glide reflections, diamonds are points of 2-fold rotational symmetry, and doubled solid lines are reflection axes. For the cm pattern, black lines denote the primitive cell boundary and the dashed lines with further spaced dashes give the centred cell. These 3 polyomino patterns correspond to the 3 patterns shown in Figure 2.6.

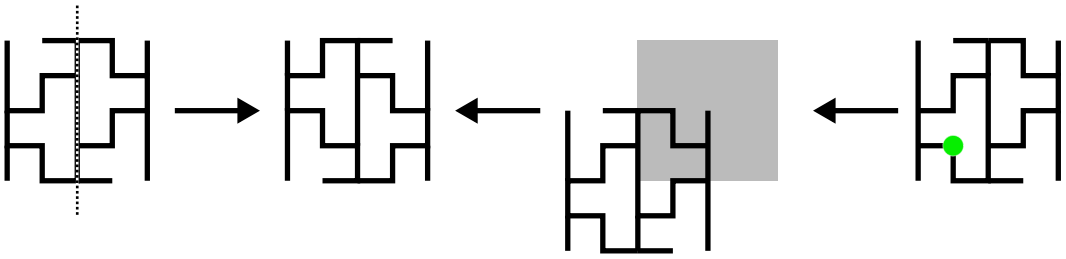


Figure 2.9: Example T-tetromino tiling in 4×4 unit cell with wallpaper group pgg. Here it is shown that a reflection (dotted line) and a 2-fold rotation (around the green circle) both map this pattern to the same new pattern, indicating that the pattern is achiral. The starting pattern is shown on the outsides, and the step with the grey square shows the pattern being wrapped around the unit cell.

rotation, but the pattern has no reflection symmetries.

In terms of wallpaper groups, this classification of chiral patterns is simple. The wallpaper groups p1, p2, p3, p4, p6 correspond to chiral patterns and the wallpaper groups pm, pg, cm, pmm, pmg, pgg, cmm, p4m, p4g, p3m1, p31m, and p6m correspond to achiral patterns. Another point about chirality is that a pattern consisting of polyominoes of only one chirality must be chiral. To see this, suppose we denote all polyominoes in the pattern as right-handed. No orientation-reversing transformation can preserve the pattern because it maps each polyomino to a left-handed polyomino, but there were no left-handed polyominoes in the original pattern. Therefore, the pattern must be chiral.

2.2.5 The Lattice Of Translational Symmetries

An important aspect of doubly-periodic patterns is the translational symmetry, which can also be called a lattice. The symmetry group of a doubly-periodic pattern contains the pure translational symmetries as a normal subgroup. Consequently, the symmetry group of any periodic polyomino pattern is a semidirect product between the translational symmetries and another group, called the point group, which is a discrete subgroup of the orthogonal group, generated by either 2-fold, 3-fold, 4-fold, or 6-fold rotations, and possibly a reflection. The choice of the semidirect product allows for 18 different ways to combine the translational symmetries with the point groups. After identifying groups that are isomorphic, this results in the 17 wallpaper groups [86].

To express the translational symmetries in a simple form, it is possible to use two translation vectors that generate the lattice in question. Denote two generating vectors as (u_x, u_y) and (v_x, v_y). The full group of translational elements is given by $\alpha(u_x, u_y) + \beta(v_x, v_y)$ where α and β are integers. However, we can choose other generating vectors for the same translational lattice by using a transformation matrix of the form

$$\begin{pmatrix} u'_x & u'_y \\ v'_x & v'_y \end{pmatrix} = \begin{pmatrix} a & b \\ c & d \end{pmatrix} \begin{pmatrix} u_x & u_y \\ v_x & v_y \end{pmatrix}, \tag{2.2}$$

where a,b,c, and d are all integers. Similar to changing basis in the real numbers, the transformation matrix must be invertible. The key is that for a matrix to be invertible in the integers, the matrix must be unimodular [87]. Figure 2.10 gives an example of two basis pairs that generate the same translational lattice; the two bases are related by

$$\begin{pmatrix} 0 & 3 \\ -2 & -1 \end{pmatrix} = \begin{pmatrix} -2 & 1 \\ 1 & -1 \end{pmatrix} \begin{pmatrix} 2 & -2 \\ 4 & -1 \end{pmatrix}. \tag{2.3}$$

Furthermore, since the transformation matrix must have determinant ± 1, the magnitude of the cross product of the basis vectors equals the same value for any choice of basis. In other words, the area of the primitive cell does not depend on the basis choice.

If we consider the wider class of integer transformation matrices with determinant nonzero (but not necessarily ± 1), we obtain the set of transformations from a lattice to one of its sublattices. A sublattice is simply a subgroup of the translational symmetry group that had defined the original lattice. An example lattice and sublattice is shown in Figure 2.11, corresponding to the equation

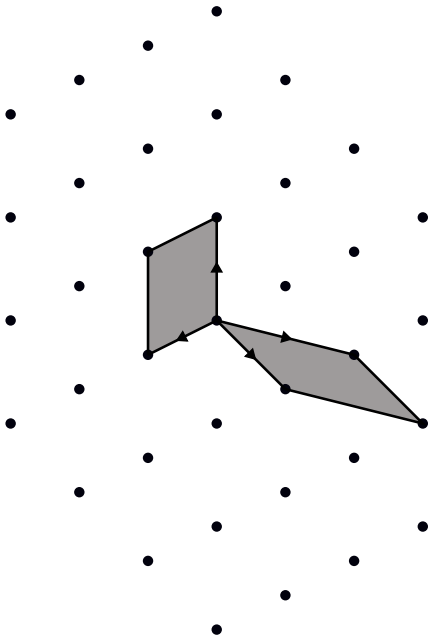


Figure 2.10: Lattice with two different pairs of generating translations shown by arrows. The primitive cell for each is also shown as a grey region. Note the area of each region is the same.

$$\begin{pmatrix} 2 & 1 \\ 0 & 2 \end{pmatrix} = \begin{pmatrix} 1 & 1 \\ 0 & 2 \end{pmatrix} \begin{pmatrix} 2 & 0 \\ 0 & 1 \end{pmatrix}. \tag{2.4}$$

The sublattice concept is quite important for understanding the structure of the lattice. However, as explained above, for every lattice (or sublattice) there is more than one choice for the basis vectors. It is important to be able to consider each of the sublattices exactly once. Therefore, it is best to use a canonical choice so that there is exactly one canonical basis choice for each sublattice. This can be done by using Hermite normal form. Intuitively, Hermite normal form assigns one of the basis vectors only one nonzero component, and the other basis vector takes position in the first octant of the plane. The basis vectors of Figure 2.11 are in Hermite normal form. To be specific, the Hermite normal form used in this thesis is given by an integer matrix

$$\begin{pmatrix} \gamma & \alpha \\ 0 & \beta \end{pmatrix}, \tag{2.5}$$

where γ and β are both greater than zero, and $0 \leq \alpha < \beta$. The Hermite normal form can be seen to be analogous to the row reduced echelon form for matrices with

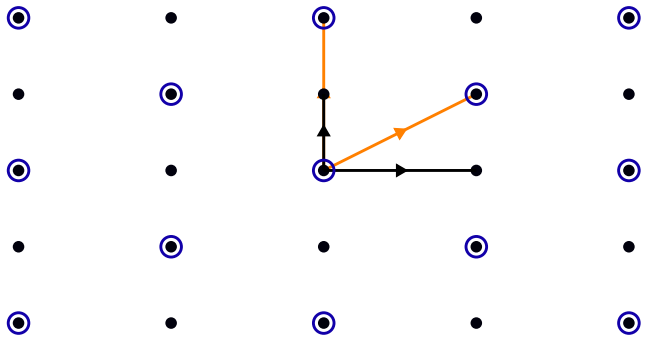


Figure 2.11: Translational lattice shown as black dots, with sublattice shown as blue circles. The dark arrows give a Hermite normal basis for the lattice, and the orange arrows give a Hermite normal basis for the sublattice.

real numbers. This means that any matrix representation of a sublattice can be transformed into Hermite normal form in a straightforward way, providing a useful way to represent translational sublattices uniquely.

2.3 Molecular Rearrangement

To model the problem of molecules that can rearrange after being initially adsorbed, we endow fully-packed polyomino systems with a dynamic moveset consisting of moves that involve rearranging a finite number of polyominoes. We will call these moves local. In the extreme case, molecules become completely fixed as soon as they bond to the substrate, leading to a configuration which is usually not fully-packed, nor ordered. If the molecules are extremely mobile while in the process of becoming adsorbed to the substrate, we can imagine the system is able to sample from the entire space of all possible molecular configurations.

However, in the intermediate case, desorption-adsorption events will only involve a few molecules at the same time. This corresponds to molecules that prefer to be adsorbed to the substrate, but can desorb and adsorb again, in a different orientation. A sequence of these local moves leads to a rearrangement of the polyomino tiling. The domino system gives the best example, since a natural set of local moves has been well explored in the literature. As mentioned in Chapter 1, domino pair flips give a natural moveset for fully-packed domino tilings.

Another kind of dynamics that has been considered in the literature are worm moves [50]. Worm moves involve taking out one domino, transforming it into two square vacancies and allowing one vacancy to diffuse until it recombines with the other vacancy, being filled again by a domino. This causes all the dominoes along

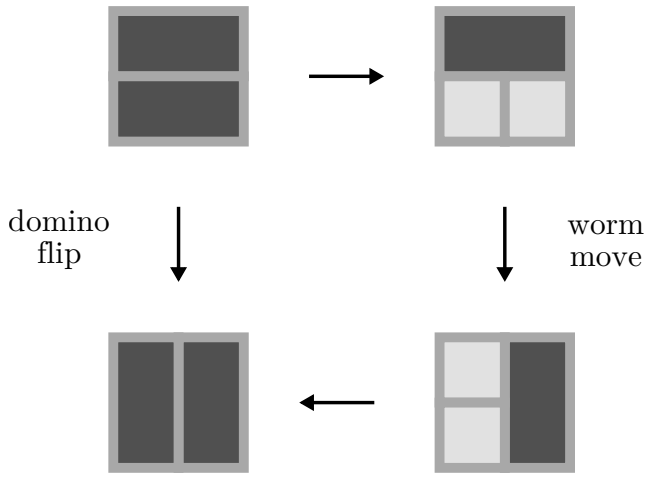


Figure 2.12: The smallest worm move, shown to be equivalent to a domino flip move. Dominoes are depicted as dark rectangles, and vacancies as light squares.

the path of the vacancy to be shifted. The domino flip move corresponds to a worm move where the vacancy only takes one step, shifting only a single pair of dominoes, as can be seen in Figure 2.12. The domino pair flip move can also be thought of as the smallest possible non-trivial monomer creation-annihilation event. A molecule is desorbed, an adjacent aligned molecule shifts to take up one of its plaquettes, then the two remaining empty plaquettes are filled by the adsorption of a new molecule. As detailed in Ref. [50], the length of these worm moves becomes suppressed at low temperatures, when considering domino-aligning interactions. When the periodic ground state is approached, the length of worm moves decreases, so that effective domino flip moves occur more frequently. Since we are investigating the ordered domino configurations, this suggests domino flip moves are a reasonable choice.

2.3.1 Domino Pair Flips

Any two domino configurations of a simply-connected region are connected by a sequence of domino pair flips. This moveset includes only two different moves; two aligned horizontal dominoes are rotated to become a vertical pair of aligned dominoes, and vice versa, as illustrated previously in Figure 1.9. Since this moveset is sufficient to rearrange any simply-connected region and our domino configuration is fully-packed on the plane, any local move can be reproduced by a sequence of domino flip moves. An example is shown in Figure 2.13. Furthermore, the two domino flip moves are the only rearrangements involving the fewest possible number of dominoes, which makes it a good candidate for the case mentioned where desorption-adsorption events only involve a small number of molecules at a time.

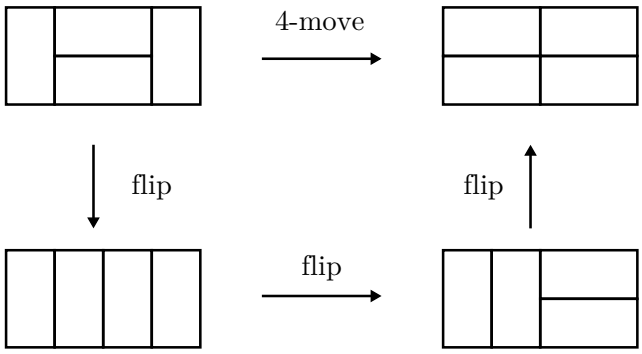


Figure 2.13: A local move that rearranges 4 dominoes simultaneously is shown, with a sequence of domino flip moves that achieves the same rearrangement.

Fully-packed domino configurations on a simply-connected region are connected to each other by sequences of domino flip moves, as mentioned already. To expand on this idea, consider the network defined such that each configuration is a node and edges exist between nodes when the corresponding configurations differ by a single flip move. We will refer to this network as the connectivity network. For a simply-connected region, there is only one connected component in the connectivity network; therefore, all configurations are accessible by starting from any one configuration, then applying domino flip moves.

However, the type of regions we use for domino packing are rectangles with periodic boundary, which have the topology of a torus, meaning they are not simply-connected. In this more general setting, some configurations are not connected by a sequence of domino flip moves. In other words, the connectivity network for domino configurations will have multiple connected components, instead of just one. Thankfully, a description of the connected components has been set out in Ref. [43], using topological techniques. A similar solution is not known for the more general polyomino packing problems, but we will make use of this specific tool for domino systems throughout this thesis.

2.3.2 Connectivity By Homology

Here we explain the method to determine whether domino configurations are connected by domino flip moves, or not, as detailed by N.C. Saldanha [43]. For the sake of simplicity, we will consider only rectangular regions of even-even dimension on which the dominoes are to be packed. It was explained by Kasteleyn [36] that two domino configurations ((a) and (b) of Figure 2.14) can be used to define a transition graph that contains double bonds and cycles (shown in Figure 2.14(c)), where

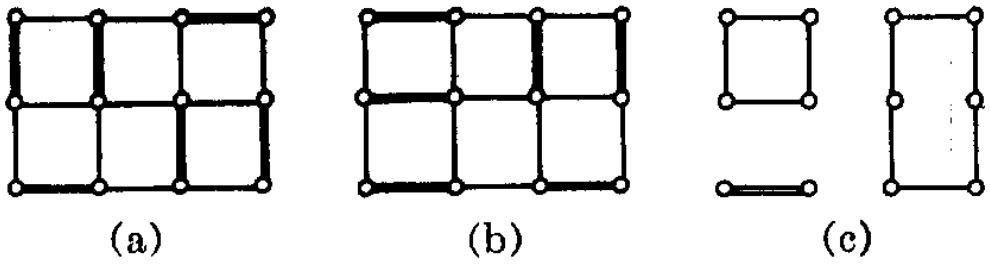


Figure 2.14: (a) and (b) Show two domino configurations (drawn as dimers), and (c) gives the corresponding transition graph, created by overlaying the two configurations. Reproduced from [36].

shifting dominoes along each of these cycles takes you from one configuration to the other. In this case, it is better to illustrate the dominoes as dimers, which we take to be edges joining the centres of the two plaquettes that define a domino, as described in Figure 2.15(b) and (c).

It is also possible to give an orientation to the cycles of the transition graph by orienting the dimers of the first configuration from dark to light squares of a superposed checkerboard, orienting the dimers of the second configuration oppositely, and then adding the two together. This gives an object that we will refer to as the oriented transition graph. An example is shown in Figure 2.16, where it is clear that the double bonds do not remain because they cancel out. This example visually suggests there might be a useful way to frame the problem of domino configuration connectivity in terms of homology, and indeed this is the method we will set out here, that was originally described in Ref. [43].

The oriented transition graph is a collection of 1-dimensional cells of the CW complex illustrated in Figure 2.15. Any such collection is known as a 1-chain and the set of 1-chains forms the 1-dimensional chain group (C_1). Similar collections of 0-cells and 2-cells are called 0-chains (C_0) and 2-chains (C_2), respectively. The boundary map associates any chain to its boundary, which is a chain of one dimension lower. Examples are shown in Figure 2.17. Overall, the chain groups and boundary maps define the chain complex containing the homological information about the problem.

Crucially, the domino pair flip moves can also be incorporated into the chain complex. Figure 2.18 shows two configurations that differ by a single flip move, and the corresponding oriented transition graph. In the oriented transition graph, it is clear that the domino flip move is the image of a 2-dimensional cell, possibly with

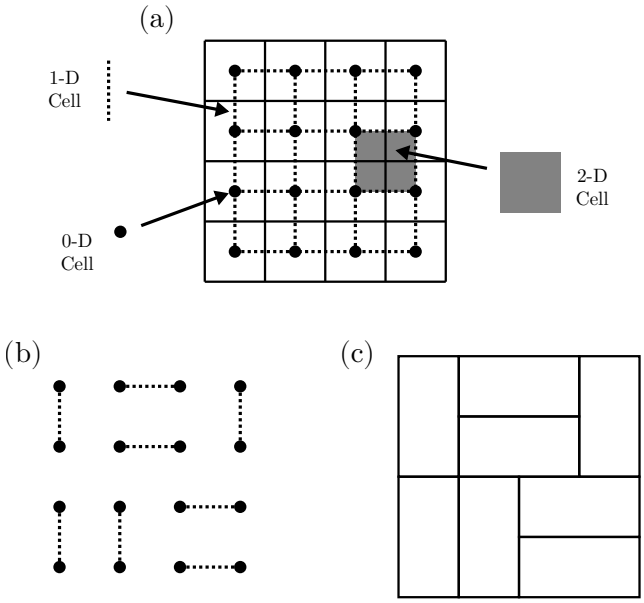


Figure 2.15: (a) Using a dimer representation, the dimers can be seen to correspond to 1-dimensional cells of a CW Complex, which is dual to the original square lattice that the dominoes reside on. Dimer (b) and Domino (c) representations are compared.

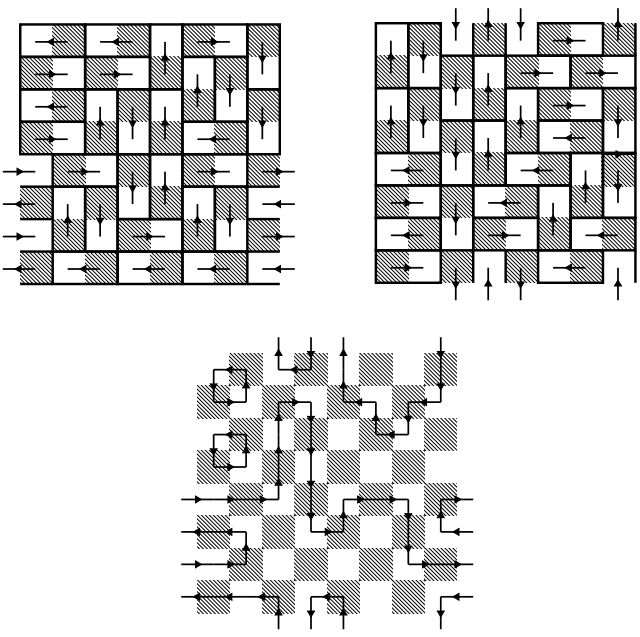


Figure 2.16: Two example configurations are shown, with dimers oriented from dark to light squares. Beneath is shown the oriented transition graph formed by the first configuration minus the second.

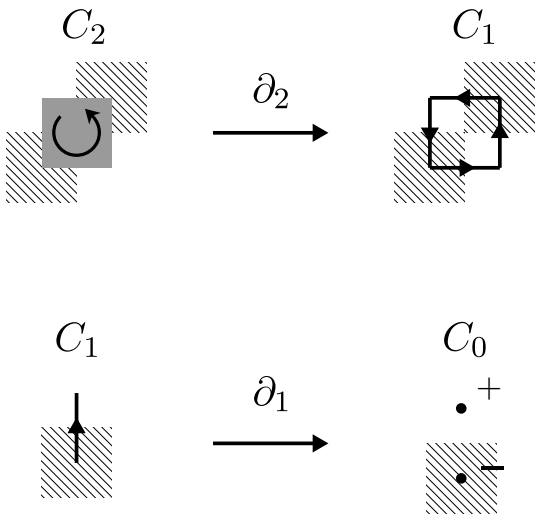


Figure 2.17: Example cells of the dimer CW Complex are shown, with the operation of the boundary map that acts on them. For each cell, the corresponding chain group (C_0, C_1, or C_2) is listed alongside.

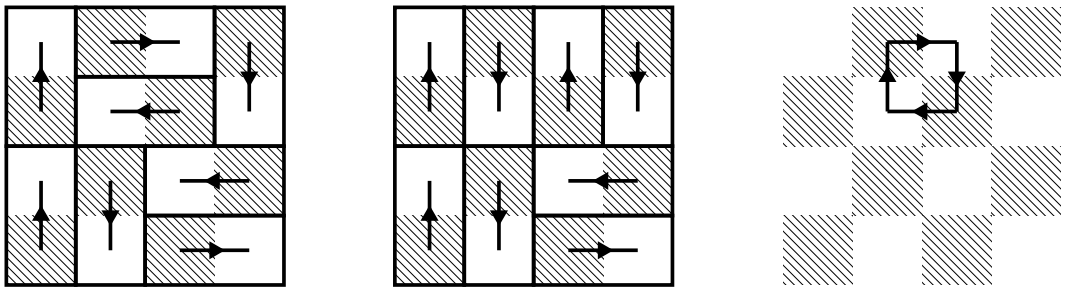


Figure 2.18: Two configurations related by a single domino pair flip move. The oriented transition graph obtained by taking the difference of the two configurations is also shown.

opposite orientation. So, the oriented transition graph in that case lies in $\mathrm{Im}(\partial_2)$. Moreover, we can say more than this, since the oriented transition graph of any two configurations connected by a sequence of domino flips must also be contained within $\mathrm{Im}(\partial_2)$. An example of this can be seen in Figure 2.19, which shows two configurations that are connected by domino flip moves.

If the oriented transition graph is not in $\mathrm{Im}(\partial_2)$, the two configurations cannot be connected by any sequence of domino flips. An example of this is shown in Figure 2.20. Furthermore, the oriented transition graph is always an element of $\ker(\partial_1)$, since it is made up of closed loops only. Therefore, by choosing one reference configuration to define an oriented transition graph for all other configurations, the quotient group $H_1 = \ker(\partial_1)/\mathrm{Im}(\partial_2)$ gives an equivalence relation between domino configurations. The exact form of each of the oriented transition graphs depends on the reference configuration, but the equivalence relation does not depend on the reference. Configurations in different equivalence classes cannot possibly be connected by a sequence of domino flip moves. Furthermore, H_1 is the homology group of our chain complex, so the equivalence classes are in fact homology classes.

2.3.3 Ladder Configurations

Domino configurations being in the same homology class is a necessary condition for being connected by a sequence of domino flips, but it is not sufficient. However, there is only one small extra condition that is required, which was proven in Ref. [43]. The condition is that the configurations being compared contain no ladders, which are diagonals covered by dominoes of only one orientation. For rectangular regions with periodic boundary, any configuration containing ladders must be completely covered by ladders and there are no domino pairs that could be flipped. We call this kind of domino configuration a ladder configuration. The set of ladder configurations can be easily enumerated, as each diagonal must contain dominoes in 1 of 2 orientations. An example configuration is shown in Figure 2.21.

Excluding ladder configurations, the set of configurations in the same homology class are exactly the set of configurations for which there exists a sequence of domino flip moves connecting them. The connectivity network of domino configurations is therefore made up of connected components containing all configurations of the same homology class, except the ladder configurations, which each exist in their own connected component, since there is no domino flip move connecting them. In the rest of this thesis, when mentioning domino configurations in the same homology class we will implicitly be excluding the ladder configurations, unless stated explicitly. This is conceptually easy to do, as ladder configurations are never in the

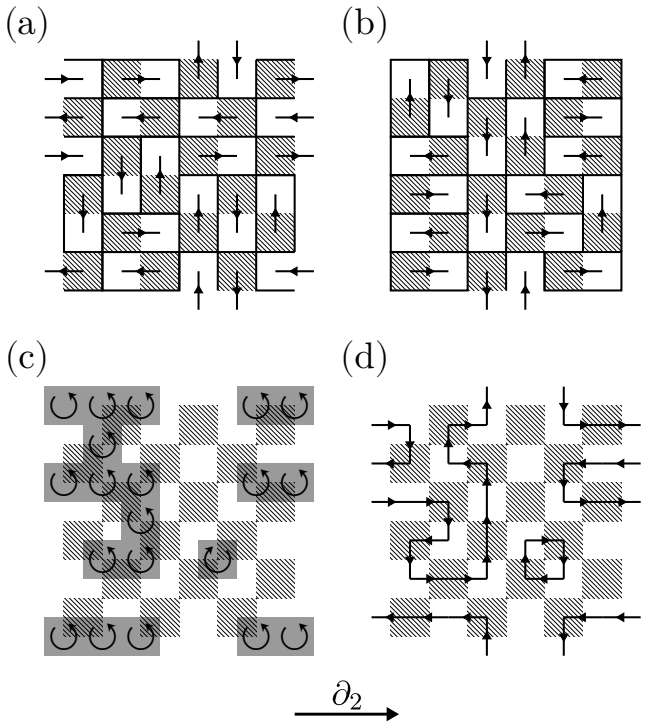


Figure 2.19: (a),(b) Two example domino configurations that are connected by domino flip moves, with (d) the oriented transition graph (an element of C_1) created by (a) minus (b). Moreover, (c) is an element of C_2, the image of which gives (d).

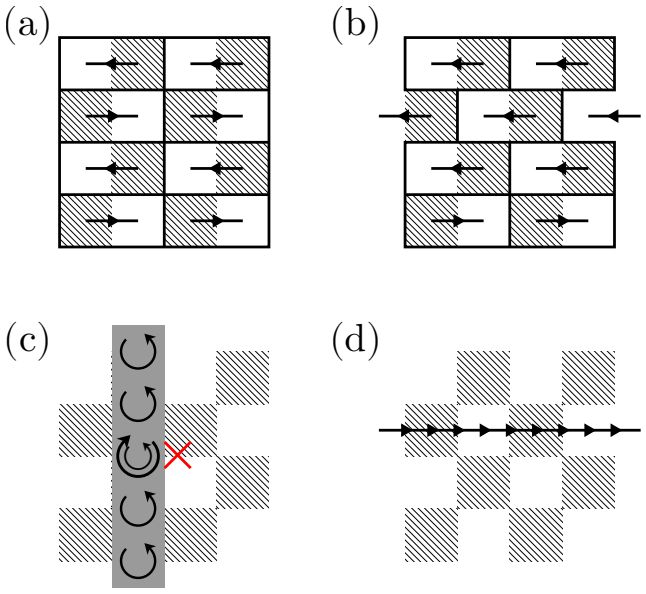


Figure 2.20: (a),(b) Show two domino configurations that are not connected by domino pair flip moves. (d) Is the corresponding oriented transition graph. (c) Shows that it is impossible to orient 2-dimensional cells (i.e. create an element of C_2) that map to the oriented transition graph.

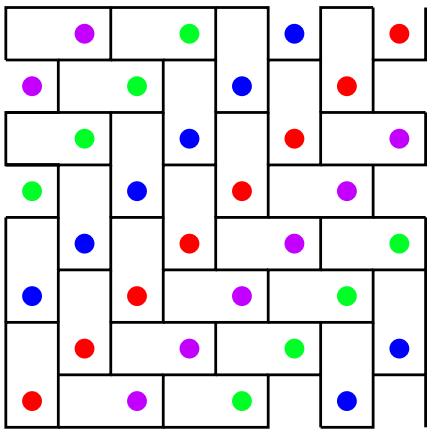


Figure 2.21: An example domino configuration made of ladders. Coloured circles indicate each of the ladders, which consist of a diagonal repeating pattern of dominoes in the same orientation.

same homology class as non-ladder configurations.

2.3.4 Connectivity Of Periodic Planar Domino Configurations

The reasons for choosing domino pair flips for defining the connectivity of configurations on a rectangular periodic region have already been explained: they are the smallest possible moves for fully-packed domino configurations, a sequence of domino flips can reproduce any other small movesets, and they are equivalent to the smallest worm moves. However, the connectivity by domino pair flips is also useful when considering periodic domino configurations on the plane. When considering the rectangular cell as being a unit cell that tiles to create a doubly-periodic domino configuration on the plane, the same connectivity criteria gives a condition for periodic patterns to be connected by a sequence of many small rearrangements that can be made at various parts of the plane to convert each part from one domino pattern to the other.

For any two periodic domino tilings of the plane, it is possible to calculate their homology class on some rectangular unit cell with periodic boundary conditions (even though this region may be larger than the area of the primitive cell of either of the two configurations). In practice, we will generate the configurations on two unit cells (possibly of different dimensions), then reproduce them both on a supercell commensurate with the dimensions of the two unit cells originally considered. This concept can be seen in Figure 2.22. In this way, when we are considering periodic ground states, our concept of connectivity by domino flips is not restricted to domino configurations that are generated on a particular unit cell with periodic boundary.

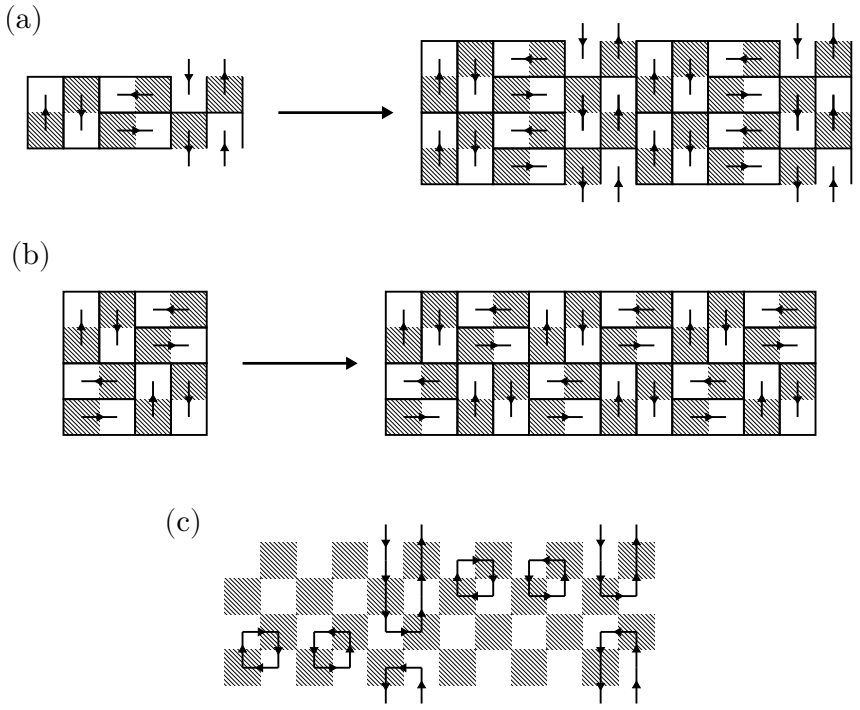


Figure 2.22: (a),(b) Two domino configurations, with representations on a larger unit cell that is commensurate with both of the original cells. (c) The oriented transition graph for the two configurations on the larger unit cell.

Consider a doubly-periodic domino configuration constructed on a unit cell, and tile the plane with it, to get the corresponding planar domino tiling. A rearrangement on the unit cell that corresponds to a sequence of domino flips can be performed on a small region of the plane, leaving the rest unchanged. However, a rearrangement on the unit cell that is inequivalent to a sequence of domino flips corresponds to a rearrangement on the plane that must involve a macroscopic number of dominoes. Figure 2.23 and Figure 2.24 give two examples for this concept. In Figure 2.23, there is no change in the homology class, and it is possible to make a transition on a small region of the plane, whereas this is not true in Figure 2.24, where the homology class changes. So, for the unit cell domino configurations with equal homology class, it would be possible to convert from one to the other planar configurations by small rearrangements at various parts of the plane independently. Indeed, this could be done by many small sequences of domino flips. Conversely, when the two unit cell domino configurations are in different homology classes, only a rearrangement involving a very large number of dominoes can be used to convert between the two configurations, which is very unlikely, as it would require a large number of molecules to desorb simultaneously.

Instead of making a macroscopic move, it would be possible to convert one region of the plane from one planar domino configuration to the other. However, when the two configurations are in different homology classes, this will inevitably disrupt other parts of the plane. This is illustrated in Figure 2.25, where the planar version of the configuration from Figure 2.20(b) is being converted into the homology-inequivalent configuration of the planar version of Figure 2.20(a). The conversion is successful at the centre of the plane, but causes a surrounding region to become even more dissimilar. We would like to have a concept of connectivity where planar configurations are only considered connected when it is possible to convert all parts of the plane from one configuration to the other, independently. This motivates us to use the homology sense of connectivity. Furthermore, it is relatively easy to calculate homology classes, which we will explain in Section 3.6, so this concept of connectivity is also practical to use.

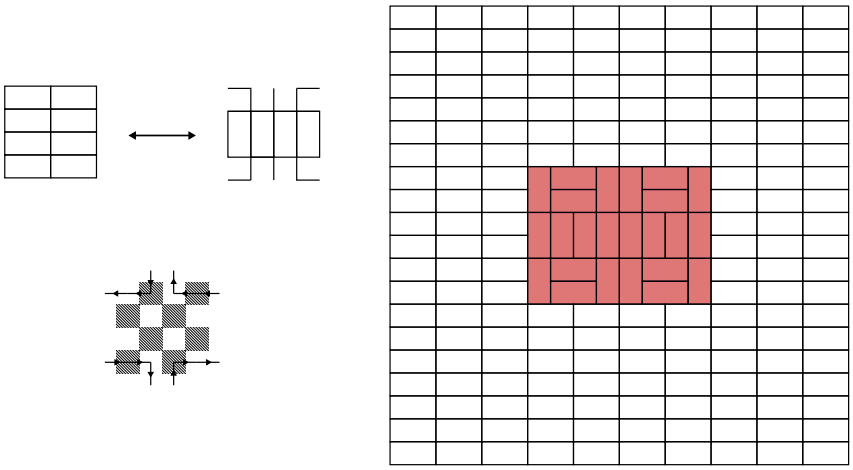


Figure 2.23: Two domino configurations on a 4×4 unit cell are shown, with their oriented transition graph, which is in the trivial homology class. In the corresponding planar tiling, it is possible to convert a finite region (highlighted in red) from one planar tiling into the other.

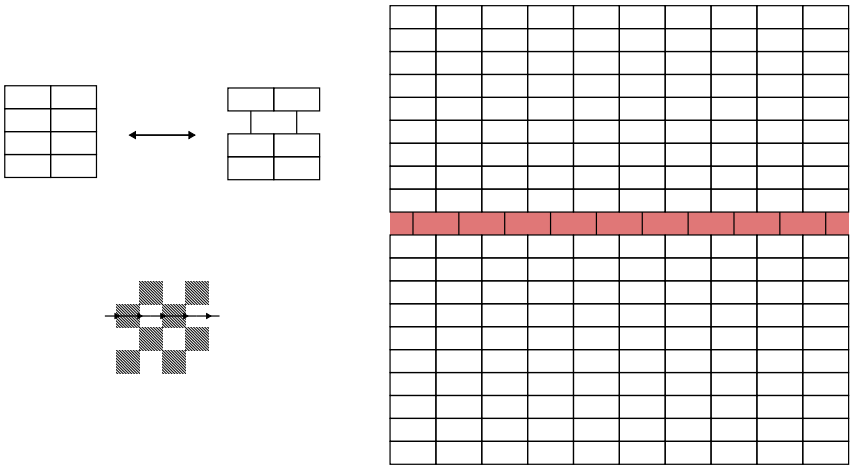


Figure 2.24: Two domino configurations on a 4×4 unit cell are shown, with their oriented transition graph, which is in a non-trivial homology class. In the corresponding planar tiling, a macroscopic region (highlighted in red) must be converted, to be able to make the kind of transition shown in the 4×4 unit cells.

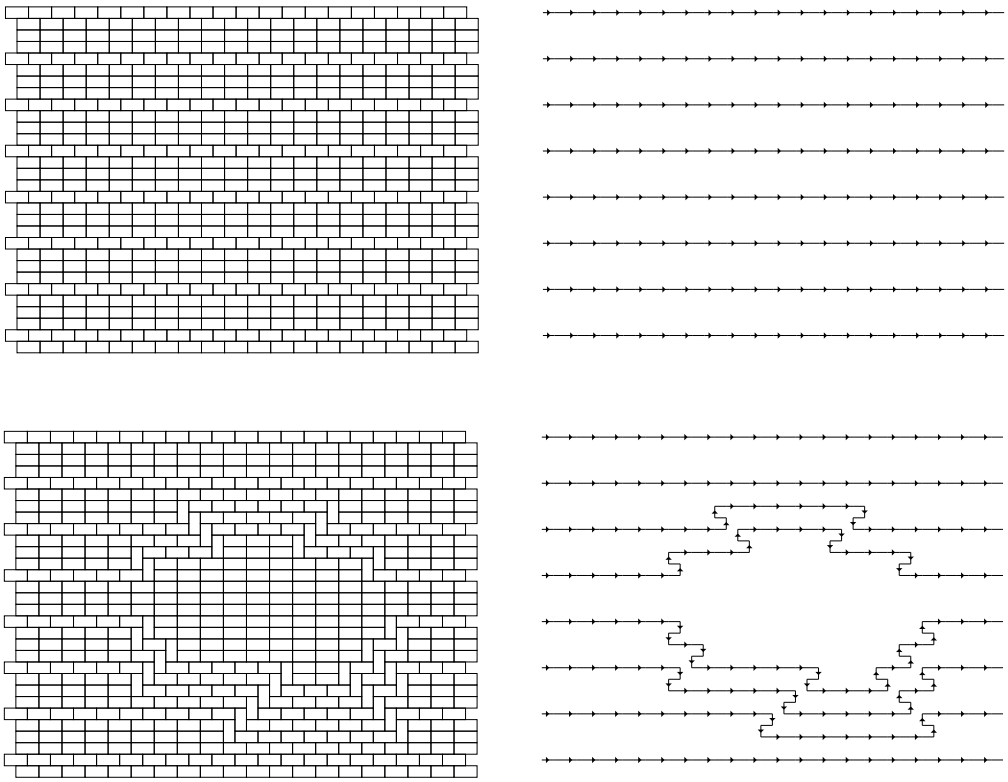


Figure 2.25: The initial configuration is shown in the top-left, and the oriented transition graph with respect to the configuration of Figure 2.20(a) is shown in the top-right. In the lower images, the center of the configuration has been converted into the all-aligned domino configuration of Figure 2.20(a) by using domino flip moves. However, since the configurations of Figure 2.20(a) and Figure 2.20(b) are in different homology classes, this inevitably causes the surrounding region to become even further from the all-aligned configuration. This can be seen topologically, as the loops of the oriented transition graph are simply being moved away from the origin, and becoming more tightly bunched in the surrounding region.

Chapter 3

Pattern Enumeration And Symmetry Finding

For the computational enumeration and analysis of polyomino tilings, we have implemented several algorithms and devised methods of our own to further this aim. In this chapter, we describe these algorithms and methods, which are important for understanding the results in the ensuing chapters.

An important part of our methodology is the enumeration of polyomino patterns. We make use of algorithm DLX from Ref. [88] to create an algorithm for enumerating patterns of general polyomino shapes. Then we outline our method for counting generated polyomino patterns so that equivalence as planar patterns is taken into account.

In Section 3.3 and Section 3.4, we describe our implementation of a symmetry-finding algorithm for polyomino configurations. This has two parts: Firstly, the main algorithm for calculating the primitive unit cell and wallpaper group. Secondly, a fast algorithm based on group theory principles that is used before the main algorithm to determine whether there are any nontrivial symmetries that need to be identified. In both parts, the algorithms benefit from a simplification of the group of possible symmetries that must be considered, by making the algorithm specific to the problem of polyomino configurations. Finally, we give detail on our specific model of interactions, use packing constraints to reduce the effective dimension of the interaction parameters, and describe the calculation for ascertaining homology classes.

3.1 Enumeration Algorithm

To be able to analyse molecular networks, we enumerate fully-packed polyomino tilings within a rectangular unit cell with periodic boundary. When referring to unit cells, we use the more general meaning, of a region that can be tiled to fill the plane with a given pattern. We use the term primitive unit cell for any unit cell of minimal area. In this thesis, we restrict to just one type of polyomino within a configuration, e.g. the right tromino and no other types. It is equally possible to use a mixture of polyomino types, although this requires greater computational resources.

The first point to this method is being able to enumerate periodic patterns with low periodicity, and thus search for the set of possible ground states. Secondly, for larger unit cells it is possible to generate an ensemble of the random tilings that appear when interactions are sufficiently weak that entropic forces dominate the energetic interactions.

We are interested in the investigation of several different polyomino shapes; therefore, the algorithm we use is suited to the general polyomino problem. Considering the general fully-packed polyomino tiling problem, each square of our underlying square lattice must be covered by a polyomino square exactly once so that polyominoes are not overlapping, and there are no gaps left over. An exception is made in Chapter 8, where we introduce fixed defects. A polyomino with a given position and orientation will cover a specific collection of squares. Therefore, if we let each square of the grid be an element, the polyominoes (with a specified position and orientation) correspond to subsets of the set of squares. To find a valid full-packed polyomino configuration, it is necessary to find a non-intersecting collection of subsets that together cover the entire set.

The allowed subsets are given by the polyominoes under consideration, at each possible position and orientation, as shown in Figure 3.1. If the size of the region is NM, the number of subsets to be specified is twice or four times this number, depending on how many different rotations the polyomino has. Therefore, the number of allowed subsets for the algorithm grows linearly with the number of elements. This seems fairly reasonable in terms of computation, but the computational effort is due to the number of ways in which these subsets can be chosen. Typically, the number of solutions grows exponentially with the number of elements. For example, with dominoes, the entropy per domino is comparable to the entropy per domino without the constraint that dominoes do not overlap [35].

When put in the above terms, the enumeration of fully-packed polyomino

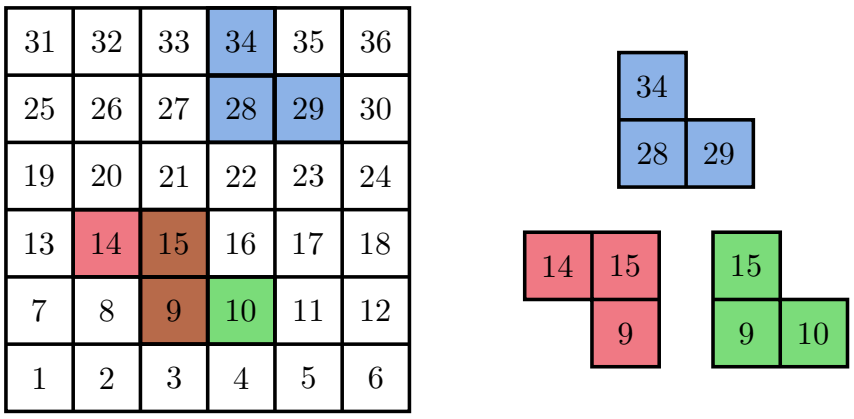


Figure 3.1: On the left, each square is an element of the set of 36 squares and is labelled by a number. On the right, 3 subsets are shown, each containing 3 elements. Also on the left, it can be seen that two of the subsets are overlapping, since they have elements in common (shown in brown).

configurations becomes the problem of enumerating all possible solutions of an exact cover problem. To solve an exact cover problem, it is possible to think of the problem abstractly. Create a matrix of ones and zeroes, with rows corresponding to the permitted subsets and columns corresponding to the elements. For each row, there must be a nonzero element in each column corresponding to an element covered by that subset. With this picture in mind, an exact cover is given by a selection of rows such that each column contains exactly one nonzero element, since this gives a collection of non-intersecting subsets that covers all elements. Figure 3.2 provides an example of this abstract matrix for right trominoes, packed in a 3×2 cell with rigid boundary.

Thankfully, there already exists an algorithm that solves the exact cover problem fairly efficiently (although, this efficiency is limited by the fact that finding an exact cover is an NP-complete problem [89]). The algorithm DLX [88] described by Donald Knuth is a depth-first algorithm that enumerates all solutions to an exact cover problem. The depth-first nature of algorithm DLX means that the algorithm can be pictured in Figure 3.3, as travelling through a search tree of partially completed solutions. The algorithm iteratively selects from the remaining subsets that do not intersect with those already used, to add subsets to a partial solution. When the algorithm cannot continue any further, it backtracks until it can continue again along a different search branch. Note that in the abstract matrix representation of the exact cover problem, this means the algorithm selects rows from the matrix to include in a partial solution matrix such that each column of the partial solution matrix is full of zeroes, except for a single element.

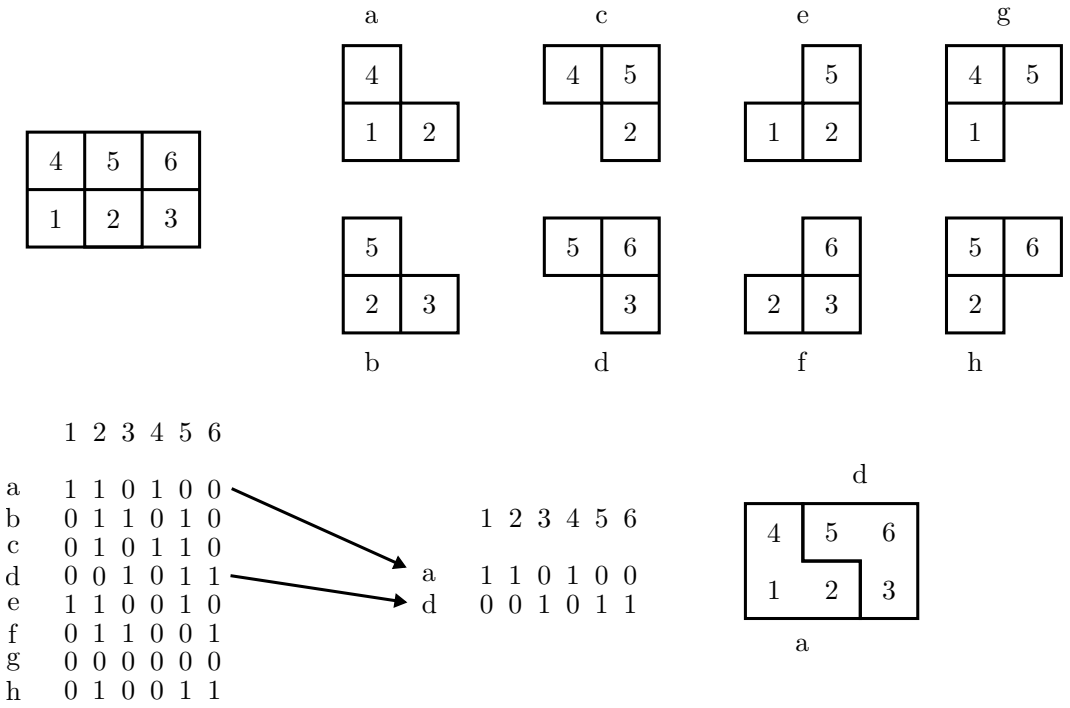


Figure 3.2: In the upper-left, a 3×2 unit cell is shown. This unit cell is to be tiled with rigid boundary by right trominoes, all possible placements shown in the upper-right. There are 8 possible placements, corresponding to letters a-h, and 6 squares to be covered; therefore, a 6×8 matrix of ones and zeroes is used to represent the abstract exact cover problem. In the lower-right, an example solution is shown, using the right trominoes labelled a and d. Note that the resulting matrix has rows from a and d, and contains exactly a single 1 per column.

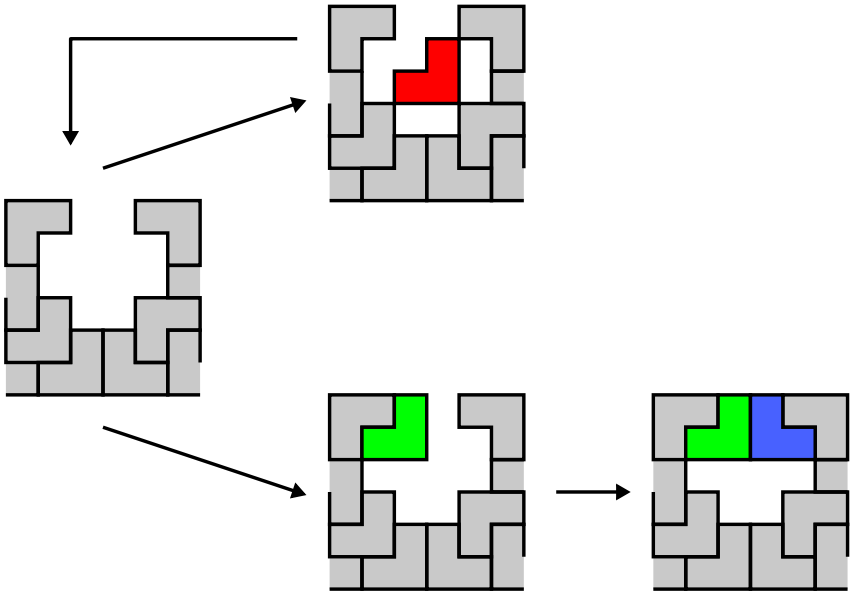


Figure 3.3: Illustration of a depth-first algorithm for right tromino packing of a 6×6 region with periodic boundary. In the upper branch, the algorithm cannot continue further, since there is a space that cannot be filled by another right tromino, so the algorithm backtracks and chooses a different branch to continue. Previously placed right trominoes are shown in grey and newly placed right trominoes are shown in colour.

For our problem, algorithm DLX uses our defined set of allowed polyominoes to enumerate all the possible polyomino configurations via depth-first search. In a naive implementation of a depth-first algorithm, it would be necessary to save the current and previous states, or keep some kind of record of which polyomino was placed and at what stage of the algorithm. The strength of algorithm DLX is that this is taken care of automatically, without any extra use of memory to keep this information, which also saves on computational time. Algorithm DLX achieves this by using pointers to set up dynamically linked lists. The network structure of these pointers contains the information about the current state of the algorithm. Data variables do not need to be directly updated. The real beauty of the algorithm is seen when going back up a step in the depth-first algorithm. This is done by rejoining pointers to a data element which was previously skipped over. This is the part that saves a great amount of time compared to a naive implementation, since no extra data structures are required to be able to make this reverse step.

Our implementation of algorithm DLX is used to enumerate polyomino configurations on a unit cell with periodic boundary conditions. By viewing the unit cell as tiling the plane, our motivation is to identify ground state energy configurations, under quite general nearest-neighbour interactions. Algorithm DLX gives an efficient way to enumerate low periodicity polyomino patterns. This has an advantage over sampling from polyomino patterns, since sampling can miss out possible low energy states. Furthermore, Markov chain Monte Carlo methods with Boltzmann acceptance rates can suffer at full packing, for example the Markov chain involving worm moves of fully-packed domino configurations, as described in Ref. [53]. By surveying lowest energy states using algorithm DLX, we can identify the kinds of ground states that might be possible in experiment.

The other reason for enumerating polyomino configurations on unit cells is to get an ensemble that can approximate the large unit cell limit statistics of random polyomino tilings. However, since algorithm DLX enumerates all configurations, this method is not great for sampling configurations on large domains (or, equivalently, with large primitive cell area), simply because there are so many to enumerate. For a given unit cell, algorithm DLX is used either to enumerate all configurations, or to prove that at least one configuration exists. However, In chapter 7, we describe a modification to algorithm DLX that allows for random sampling of polyomino configurations.

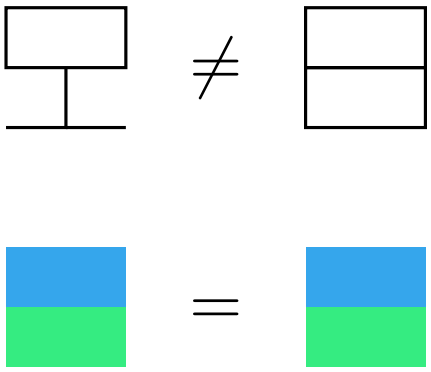


Figure 3.4: Two domino configurations on a 2×2 unit cell. When drawn as edges, it is clear that they are two different configurations, but when drawn as squares with colour corresponding to the domino they belong to, the two configurations become indistinguishable.

3.1.1 Polyomino Representation As Points On Periodic Grids

We have explained here that algorithm DLX can be put into practice by considering the squares of our grid as elements and the polyominoes as subsets comprising several elements. However, for our implementation of algorithm DLX, we use points along the edges of the polyominoes to represent the configuration. This is for two reasons. Firstly, when polyominoes are allowed to wrap around the unit cell boundary, several different polyomino configurations become indistinguishable when representing them purely as collections of squares, since the edges of the polyomino are important, as shown in Figure 3.4. Secondly, our symmetry finding algorithm works by using points as input, so by using points directly in algorithm DLX, we can avoid the step of converting polyominoes as represented by their squares, into polyominoes as represented by points along their edges.

To represent polyominoes by the points along their edges, we use a grid twice as fine as the square grid that the polyominoes actually lie on. We call this the double grid. The vertices of the double grid are filled by a point whenever there is a polyomino edge lying over that point, and unfilled otherwise. An example configuration is given in Figure 3.5. Furthermore, we can assign all points on lower and left edges of a polyomino to belong to that polyomino. In this way, each point is assigned to exactly one polyomino so that a single polyomino can be represented as points, for example the right trominoes in Figure 3.6.

In terms of algorithm DLX, we are using vertices of the double grid as the elements to be covered, and the permitted subsets are of the kind shown in Figure 3.6. However, we must make a modification to algorithm DLX because we don't

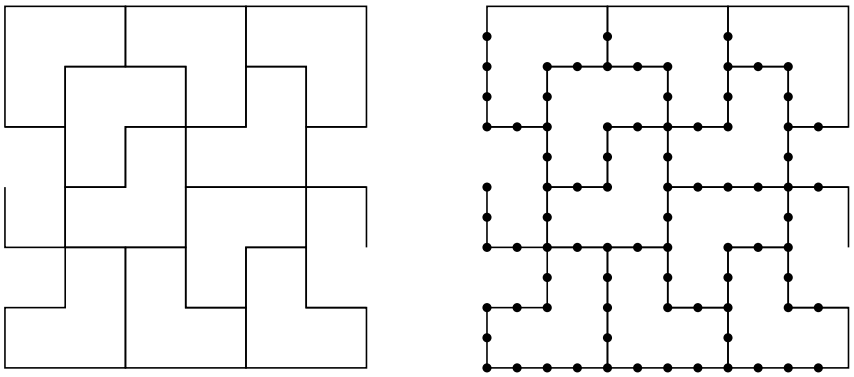


Figure 3.5: A fully-packed right tromino configuration on a 6×6 unit cell, with corresponding representation as points.

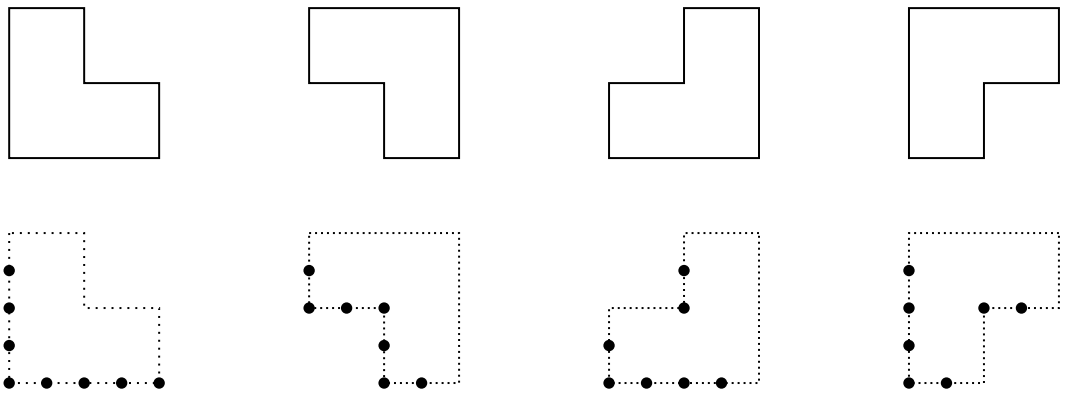


Figure 3.6: The right trominoes, in the representation by points. Note that there are only points on the lower and left edges, to avoid overlapping points in a polyomino configuration.

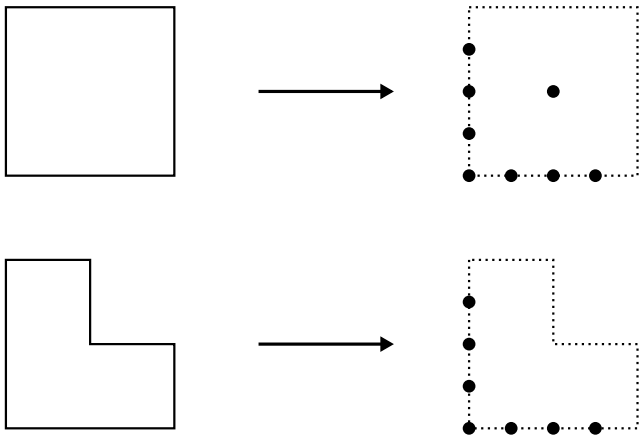


Figure 3.7: Square tetromino and right tromino, with corresponding representation by points. The dashed line is not explicitly recorded, but is implied by the filled points that coincide with the original square grid (primary points).

need all vertices to be covered. To be able to modify algorithm DLX, we make a distinction between vertex types. There are vertices in the same location as vertices of the original grid (call this type primary), and there are vertices that lie directly between them (call this type secondary). Other vertices of the double grid are not needed, since they can never be covered by polyomino edges. With this terminology, the primary vertices can be seen to indicate that the square to the upper-right is covered by the polyomino, and the secondary vertices can be seen to signal that an edge of the polyomino lies over that position.

In any fully-packed polyomino configuration, all primary vertices must be covered, and the secondary vertices are optional. Thankfully, algorithm DLX as described in Ref. [88] allows for the possibility of optional elements, which can be covered at most once. This is exactly what is needed for our secondary vertices.

Since we are using filled primary vertices as signifying that the square to the upper-right is occupied by the polyomino, this will result in some cases where there are points at locations where there is no polyomino edge. For example, the square tetromino will have a primary filled point in the interior, as shown in Figure 3.7. However, this is of no detriment to the algorithm, and also means that all polyominoes are clearly distinguished, for example a single right tromino and square tetromino are now distinguished, also shown in Figure 3.7.

Using our representation of polyominoes as points of a double grid provides a sufficient polyomino representation. By drawing straight lines between the primary points that have a secondary point immediately between them, we unambiguously obtain a pattern made up of straight edges, which is the actual polyomino configu-

ration.

To implement algorithm DLX, we define necessary elements, optional elements, and the allowed subsets of the exact cover in the form of a doubly-linked list. Algorithm DLX can then be applied, which is detailed in Ref. [88]. The algorithm acts recursively, able to call itself to complete more of the partial solution, or return to where the previous call was made, to undo the previous action. When every primary vertex has been covered, a solution has been found. The direct output of Algorithm DLX is a list of pointers to each of the rows corresponding to polyominoes that make up the solution. This identifies the configuration. From here, we encode the configuration as a matrix $Grid(i, j)$ with values of 1 and 0, value 1 corresponding to a point in the representation of polyomino configuration as points. We also give values (always 0) to the positions that cannot be covered, unlike in algorithm DLX, to simplify the symmetry finding algorithms that come later.

At this stage of the algorithm, we analyse the configuration to find its symmetry group, area of primitive cell, count of interactions, and other properties of interest, as well as printing the interesting configurations as PNG files. This analysis takes a significant amount of the computational time of the algorithm; the algorithm DLX is quite efficient, but to calculate properties of many configurations takes up computational resources regardless of the efficiency with which they are generated.

3.2 Equivalent Configurations

3.2.1 Principles For Equivalence

We have implemented algorithm DLX to generate configurations on a unit cell. Tiling the plane with this unit cell, we obtain a doubly-periodic planar pattern that can be thought of as a possible ground state. However, when thinking of polyomino patterns of the plane, there will generally be several configurations on the unit cell that correspond to the same planar tiling. An example of two configurations on a 3×3 unit cell that lead to the same planar tiling is shown in Figure 3.8.

We consider planar tilings that are related by translational or rotational symmetry to be equivalent. In other words, when viewing a planar tiling, we don't pick out any particular point as an origin of the plane. We think of rotated planar tilings as being equivalent because there is a physical rotation of the molecular network around the vertical axis that relates the configurations. An example is shown in Figure 3.9. However, for two planar tilings that can only be related by orientation reversing symmetry, we consider them to be inequivalent. This is because there is no simple in-plane rigid transformation between them. Furthermore, we have

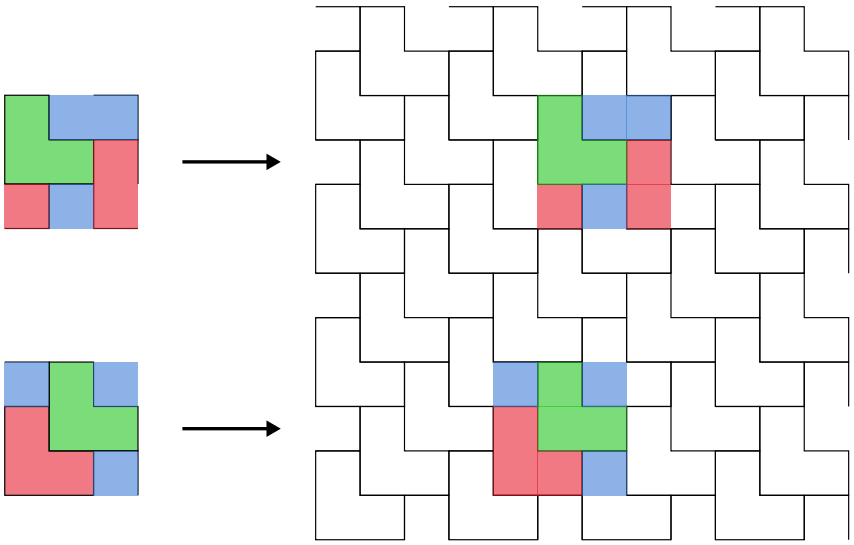


Figure 3.8: Two configurations of right trominoes on 3×3 unit cells, which both correspond to the same planar tiling. We think of these configurations as being different on their unit cell, yet the same on the plane. The 3 colours signify the 3 right trominoes of the unit cell.

defined our polyominoes as being one-sided in Figure 1.14, so it is natural to define the planar tilings with a similar equivalence relation.

3.2.2 Configuration Counting

One way to properly account for the polyomino configurations that we would like to think of as equivalent is to record representative polyomino configurations in a list and use it to compare with newly generated configurations. However, this would require many comparisons, each involving many tests of Euclidean transformations.

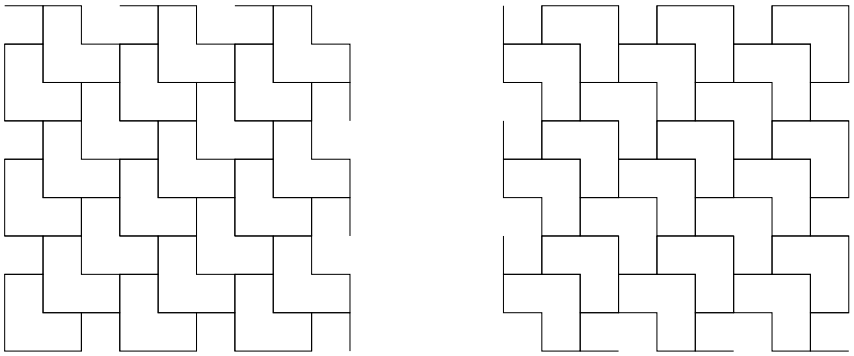


Figure 3.9: Two planar patterns of right trominoes that we consider to be equivalent, due to the rotational symmetry that maps one to the other.

This would result in a prohibitive computational cost. Therefore, rather than explicitly testing whether configurations are equivalent as planar patterns, we use a modified increment for counting configurations, which effectively takes into account the equivalent configurations. This means the analysis of polyomino configurations can be done in terms of the corresponding planar tilings. To properly take account of the equivalence of configurations, we count configurations on the unit cell by a value inversely proportional to the multiplicity of configurations generated on the unit cell that we would consider equivalent as planar tilings.

Since we calculate the symmetries of each configuration, this information can be used to determine the number of configurations that are equivalent as planar tilings. Given a configuration on a unit cell, all other configurations that are related by translational symmetry are equivalent. The number of configurations on the unit cell that are equivalent by translation is simply equal to the area of the primitive cell. This is because the polyomino configuration can only be shifted by integer translations to obtain another valid configuration on the underlying grid, and each translation vector that is outside of the primitive cell maps to a configuration that can be found by a translation vector that lies within the primitive cell. Also, each translation vector that lies within the primitive cell transforms the configuration to a different configuration, by virtue of the definition of the primitive cell. Another way to picture this is by imagining all the possible positions to choose the origin of the unit cell that will lead to different configurations on the unit cell, for the same planar pattern. This is the set of integer points within the primitive cell, shown in Figure 3.10.

For each translation-equivalent configuration on the unit cell, there will generally be several rotation-equivalent configurations. The number of equivalent configurations depends on the rotational symmetry of the pattern. Polyomino patterns can have 1-fold, 2-fold, or 4-fold rotational symmetry, determined by the point of greatest rotational symmetry in the pattern. 3-Fold or 6-fold rotational symmetry is impossible because polyominoes themselves are made of square blocks joined together. We use the term 1-fold rotational symmetry to denote patterns without any rotational symmetry and introduce the symbol R for the order of rotational symmetry, which can take values 1,2, or 4, corresponding to 1-fold, 2-fold, or 4-fold symmetric patterns.

If the polyomino pattern has 4-fold rotational symmetry, any rotation is equivalent to a translation, so there are no extra equivalent configurations. If the pattern has 2-fold rotational symmetry, rotation by π is equivalent to translation, but rotation by $\pi/2$ is not, so the number of equivalent rotations is double the

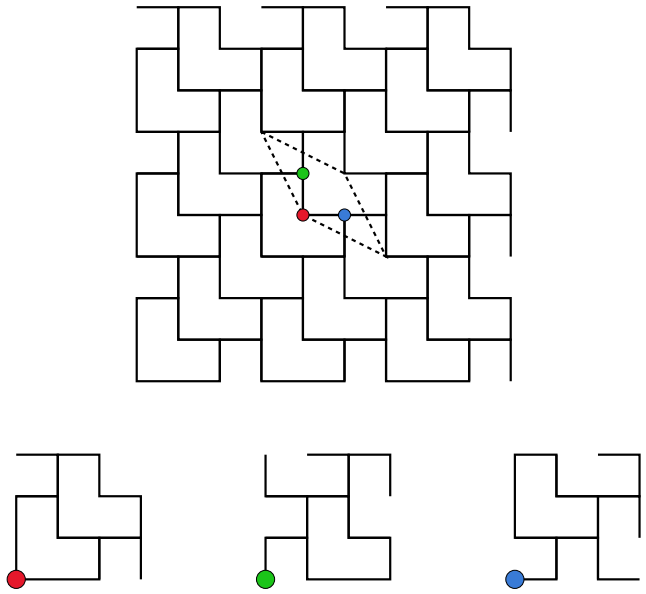


Figure 3.10: A right tromino pattern is shown with its 3 translation equivalent configurations on the 3×3 unit cell. The lower-left corner of each of their unit cells corresponds to the 3 integer positions within the primitive cell of the pattern. The 3 positions are shown as coloured dots, and the primitive cell is shown as a dashed outline.

number of translation-equivalent configurations. Finally, if the pattern has 1-fold rotational symmetry, no rotations are equivalent to translations; therefore, the number of equivalent configurations is four times the number of translation-equivalent configurations.

Considering the above points, and denoting the area of the primitive cell as A, the number of configurations that are equivalent to a planar pattern with primitive cell area A and rotational order R is $4A/R$. We show all equivalent configurations for one specific example in Figure 3.11. Now that we have the number of equivalent configurations, we can use the inverse value $R/(4A)$ to count configurations for the analysis of various properties of the planar patterns. In particular, if there is only one planar pattern with a desired property, we generate $4A/R$ equivalent configurations of this on the unit cell. By counting each of them by $R/(4A)$, we find there to be exactly one tiling with the desired property, once we finish the enumeration of the unit cell.

There is one extra complication for this method of counting configurations. For configurations related by translations or rotations by π, it is always possible to generate equivalent configurations on the same rectangular unit cell. However, this is not always true for rotations by $\pi/2$. This is shown in the example of Figure 3.12.

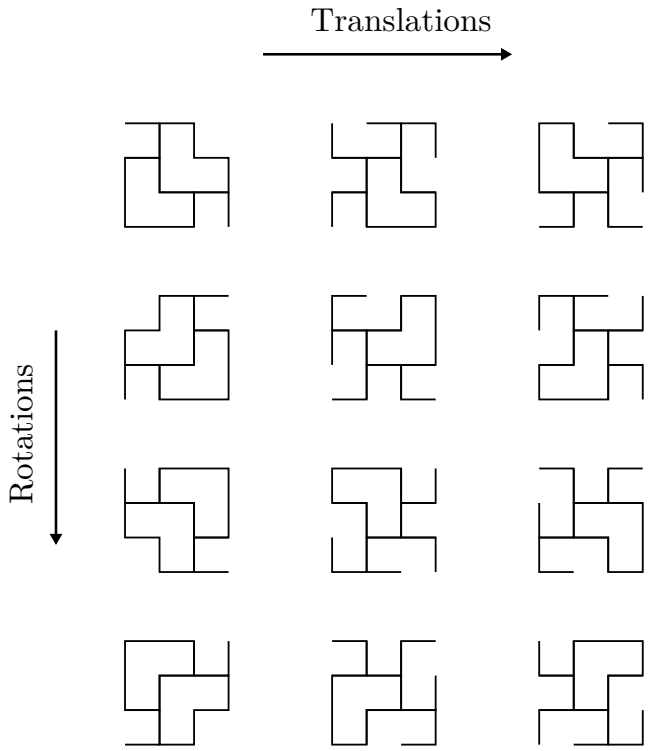


Figure 3.11: All 12 equivalent configurations of an example pattern of right trominoes are shown on the 3×3 unit cell. In this case, the area of the primitive unit cell is 3, so there are three translation-equivalent configurations. Also, the pattern has no rotational symmetry, giving four rotation-equivalent configurations for each translation.

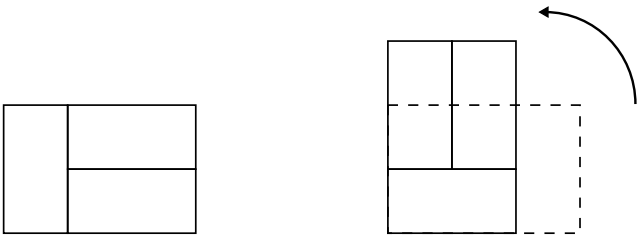


Figure 3.12: A domino configuration on the 3×2 unit cell. In this case, the equivalent configuration that is rotated by $\pi/2$ cannot be generated by the unit cell. Another horizontal domino cannot fit in the lower-right of the unit cell, so the rotation-equivalent configuration cannot be generated.

To be able to count configurations on rectangular unit cells, it is necessary to be able to decide whether the $\pi/2$ rotated version of a given configuration can be generated on the same unit cell. We describe a simple algorithm for this problem in the rest of this section.

The translations of any polyomino configuration form a sublattice of the translations of the underlying square grid. This is fairly intuitive, but the full explanation is detailed in Subsection 3.3.1. Because of this, we are always able to suppose the basis vectors for the translational symmetry of a configuration are given by the rows of the integer matrix

$$\begin{pmatrix} \gamma & \alpha \\ 0 & \beta \end{pmatrix}, \tag{3.1}$$

which is in Hermite normal form. Furthermore, suppose the configuration has been generated on the $N \times M$ unit cell. This means the $(N, 0)$ and $(0, M)$ translation vectors form a sublattice of the lattice of translations of the configuration. In other words, it is possible to find integers a,b,c,d that satisfy the following equation

$$\begin{pmatrix} N & 0 \\ 0 & M \end{pmatrix} = \begin{pmatrix} a & b \\ c & d \end{pmatrix} \begin{pmatrix} \gamma & \alpha \\ 0 & \beta \end{pmatrix}. \tag{3.2}$$

From this matrix equation, it can be calculated that α,β, and γ must obey

$$\begin{gathered} \gamma \frac{\beta}{\gcd(\alpha, \beta)} \mid N, \\ \beta \mid M, \end{gathered} \tag{3.3}$$

where gcd is the greatest common divisor. Furthermore, if the $\pi/2$ rotated configuration can also be generated on this unit cell, it means that we also have the conditions

$$\begin{aligned} \beta &\mid N, \\ \gamma\frac{\beta}{\gcd(\alpha,\beta)} &\mid M. \end{aligned} \tag{3.4}$$

Combining these four conditions, we can reduce them to the two conditions

$$\begin{aligned} \frac{\gamma\beta}{\gcd(\gamma,\alpha,\beta)} &\mid N, \\ \frac{\gamma\beta}{\gcd(\gamma,\alpha,\beta)} &\mid M. \end{aligned} \tag{3.5}$$

But in fact, this is exactly the condition that there exists a square sublattice of the translational lattice of the configuration, and that this square sublattice has the unit cell as a sublattice. This can be seen by calculating the smallest square sublattice of the translational lattice of the configuration. By smallest, we mean having smallest translations. After a short calculation (described in Section 3.4.2), the smallest square sublattice can be found to be generated by the rows of the matrix

$$\begin{pmatrix} \frac{\gamma\beta}{\gcd(\gamma,\alpha,\beta)} & 0 \\ 0 & \frac{\gamma\beta}{\gcd(\gamma,\alpha,\beta)} \end{pmatrix}. \tag{3.6}$$

The conditions in Equation (3.5) hold true exactly when this square sublattice has the unit cell as a sublattice. Therefore, this is also the necessary and sufficient condition for a $\pi/2$ rotated version of a configuration to be able to be generated on the same unit cell. As an example, the configuration shown in Figure 3.12 has generating translation vectors $(3,0)$ and $(0,2)$, which are in Hermite normal form; consequently, $\gamma = 3$, $\beta = 2$, and $\alpha = 0$. Using Equation (3.6), the smallest square sublattice has sides $(6,0)$ and $(0,6)$. The unit cell with sides $(3,0)$ and $(0,2)$ clearly does not contain the square of sides $(6,0)$ and $(0,6)$; therefore, the $\pi/2$ rotated configuration cannot be generated on the 3×2 unit cell. Conversely, any unit cell that is formed by tiling the 6×6 unit cell can generate the $\pi/2$ rotated configuration.

For our implementation, rather than explicitly finding the Hermite normal form of the translational symmetries and calculating Equation (3.6), we can make a quick test. Any square lattice that has the unit cell as a sublattice must also have the square lattice generated by $(\gcd(N,M),0))$ and $(0,\gcd(N,M))$ as a sublattice. Also, the square lattice generated by $(\gcd(N,M),0)$ and $(0,\gcd(N,M))$ always has the unit cell as a sublattice. Therefore, the condition stated above is equivalent to the condition that the translations $(\gcd(N,M),0))$ and $(0,\gcd(N,M))$ are symmetries of the configuration.

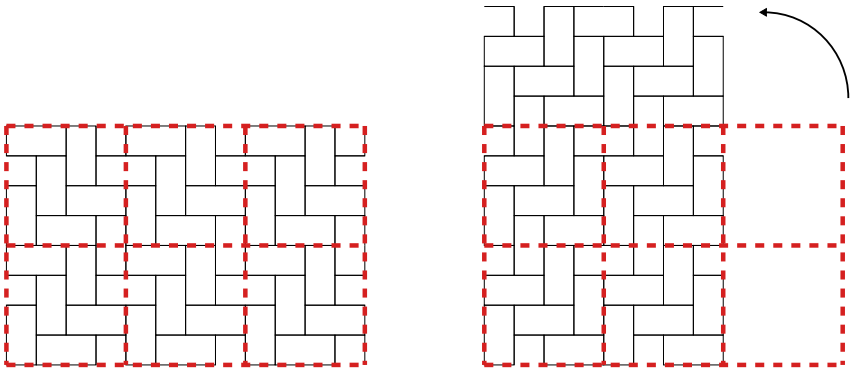


Figure 3.13: A domino configuration on the 12×8 unit cell. In this case, the configuration has the square lattice generated by $(4, 0)$ and $(0, 4)$ as a translational subgroup. This square lattice also has the lattice generated by $(12, 0)$ and $(0, 8)$ as a translational subgroup, therefore it can be seen to tile the unit cell, with the red dashed lines. The rotated configuration can be generated on this unit cell because of this.

The square with sides $(\gcd(N, M), 0)$ and $(0, \gcd(N, M))$ is the largest square that tiles the unit cell. This gives an intuitive explanation for our method. When the configuration can be split into squares that tile the unit cell, it is possible to generate the rotation-equivalent configuration, since each square can be imagined as rotating by $\pi/2$ around its centre to generate the new configuration. This is depicted in Figure 3.13.

The two symmetries $(\gcd(N, M), 0)$ and $(0, \gcd(N, M))$ are easily tested, and this information is used to determine the value that should be used for counting the generated configuration. Table 3.1 details the possible increments, depending on the rotational symmetries of the configuration and whether the rotated configuration can be formed. By using these counting increments, we are able to account for the planar-equivalent configurations that are generated on the unit cell.

3.3 Symmetry Library

Having generated configurations on a unit cell and related them to planar patterns, we can analyse the symmetries to identify the wallpaper group and primitive cell of the pattern. As mentioned previously, the wallpaper group gives a natural categorisation for the tiling and contains information about the symmetries. Another important aspect of the symmetry group is the subgroup of translations. In this section we describe our method for finding the wallpaper group and lattice of translational symmetries.

rotated possible	rotational order	counting increment
1	1	1/4
1	2	1/2
1	4	1
0	1	1/2
0	2	1

Table 3.1: The various possible counting increments are shown, given the rotational order of the configuration, and the parameter called rotated possible, which takes value 1 when the $\pi/2$ rotated configuration can be generated within the same unit cell. The combination of rotated possible = 0 and rotational order = 4 does not occur because the existance of a 4-fold rotation implies the $\pi/2$ rotated configuration is equivalent to translation, meaning it can be generated.

We analyse symmetries by implementing an algorithm based on the principles outlined in Refs. [83, 86, 90]. Our algorithm has an advantage over off-the-shelf symmetry detection algorithms because we make use of the fact that we are generating configurations made of fully-packed polyominoes, which lie on an underlying square grid. Therefore, we only need to investigate a finite number of discrete symmetries rather than a continuous set of symmetries, which makes great savings on computational effort and avoids the need to set tolerances for finding the symmetries - we can check exactly.

Furthermore, we analyse symmetries in an efficient order, making use of knowledge gained to eliminate the possibility of other symmetries, which means that not all symmetries need to be checked explicitly. Our method of checking symmetries one at a time to reach a final conclusion about the wallpaper group was inspired in particular by the work in [90]; however, the representation of polyominoes, the symmetry checks we make, and the structure of the flow diagram are distinct components of our algorithm.

3.3.1 Symmetries Of Polyomino Patterns

We have claimed that our algorithm only needs to test for a discrete set of symmetries. Here, we give details of this symmetry group, which contains the symmetry group of any polyomino pattern as a subgroup. A polyomino pattern is made up of a collection of polyominoes, which can be visually specified by the boundaries of all polyominoes in the pattern. The boundaries of polyominoes can be thought of as being made up of a collection of straight edges that coincide with an underlying square grid, and thus have unit length and integer-valued coordinates as end points.

The full set of transformations under consideration are the Euclidean trans-

formations described in Subsection 2.2.1. If one of those transformations preserves a polyomino configuration, it must map at least one edge to another edge (except in the trivial case of a polyomino configuration with zero polyominoes). The only rotation matrix that maps an edge of the square grid to another edge must have a rotation angle with one of the values 0, $\pi/2$, π, or $3\pi/2$, since it can only be mapped to a horizontal or vertical edge. Furthermore, the translation vector of the transformation must have both components integers, otherwise the endpoints of the edge cannot be mapped to endpoints of another edge. Therefore, the possible group of symmetries that preserve polyomino configurations are of the form given by

$$\begin{pmatrix} x \\ y \end{pmatrix} \rightarrow \begin{pmatrix} a & b \\ c & d \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} + \begin{pmatrix} z \\ w \end{pmatrix}, \tag{3.7}$$

where the matrix defined by a,b,c, and d is one of the signed permutation matrices, and z,w are integers. An example transformation is shown in Figure 3.14. In fact, this group of possible symmetry transformations is the symmetry group of the underlying grid. Therefore, the most symmetric polyomino configuration has a monomer on every square. The symmetry group of any other polyomino configuration must be a subgroup of this. In other words, any symmetry operation that preserves a polyomino configuration must also preserve the underlying square grid. This is not always the case for tiling problems, as illustrated in Figure 3.15. Furthermore, our polyomino configurations are generated on an $N \times M$ unit cell; therefore, there are only a finite set of symmetries that need to be considered, in addition to the translations of the unit cell. Because of this, we are able to design a symmetry finding algorithm that only needs to search through a discrete, finite set of possible symmetries.

We encode the polyomino configuration as a set of points on a double grid, as explained in Subsection 3.1.1. We must consider whether this representation is sufficient for comparing configurations by Euclidean transformations. To make this easier, we will only consider the transformations that preserve the underlying square grid, which include all possible symmetries of polyomino patterns, as just explained. It is clear that if there is a transformation between two polyomino configurations, the same transformation maps one configuration to the other in the representation as points on a double grid, since our representation can be seen as a subset of the continuum of points that form the boundary of polyominoes. However, we also require that when two configurations in the points representation are related by a transformation, the true polyomino configurations are also related by that transformation.

$$\begin{pmatrix} x \\ y \end{pmatrix} \rightarrow \begin{pmatrix} 0 & -1 \\ 1 & 0 \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} + \begin{pmatrix} 3 \\ 5 \end{pmatrix}$$

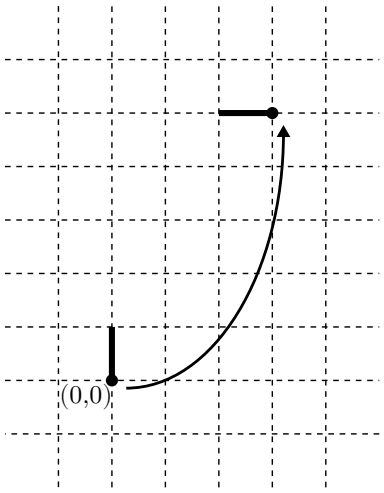


Figure 3.14: An example Euclidean transformation that maps one edge of the square grid to another. The equation for the transformation is shown alongside.

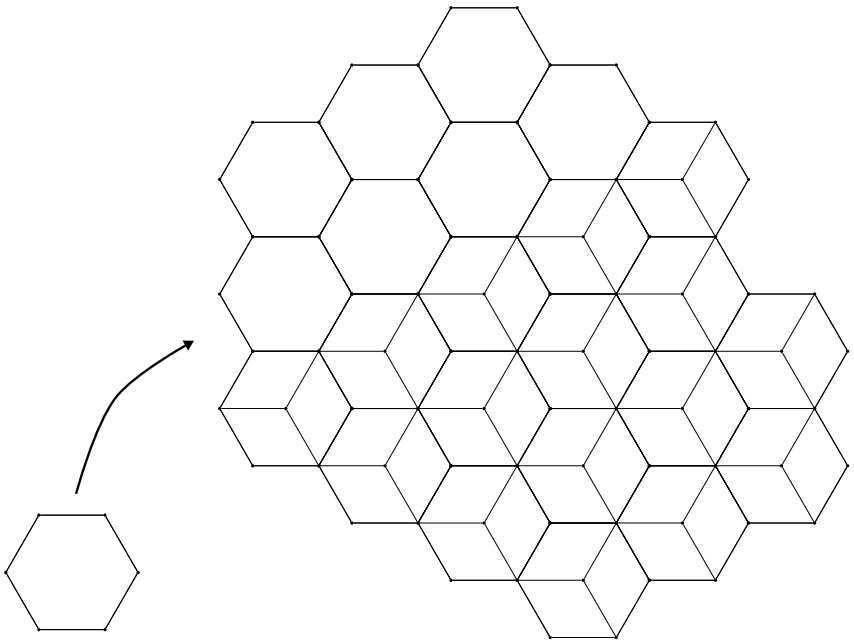


Figure 3.15: Placing hexagonal tiles onto a grid of rhombi creates points of 6-fold symmetry where there were none before. Therefore, the symmetry of the tiling is not a subgroup of the underlying grid symmetry.

Again, we can consider breaking the boundary of polyominoes into edges of unit length. In the points representation, an edge occurs wherever a point is at an odd position on the double grid. Any transformation that is an element of the symmetries of the square grid will map a point at an odd position to another point at an odd position. Therefore, in the corresponding polyomino configuration, an edge is being mapped to another edge. Furthermore, the point in odd position on the square grid must have adjacent points at only two positions. This provides an axis of orientation that will correspond to the orientation of the edge being represented, so the information of the edge orientation is taken into account by the points representation. In conclusion, the points representation is sufficient for testing the symmetries of the true polyomino configuration, which justifies our use of this representation.

3.3.2 Symmetry Checks

To test whether a symmetry operation preserves the configuration, the operation is applied, then the configuration is wrapped around the unit cell by applying mod(N) and mod(M) to the x and y components, as explained in Subsection 2.2.2. If the configuration remains unchanged, the symmetry is obeyed. For our representation as points, this can be implemented straightforwardly. Each filled point must be mapped to another filled point, otherwise the symmetry does not preserve the polyomino configuration. For a given transformation operation, we will refer to this process as testing for symmetry.

Using the above-mentioned representation and method to test for symmetry, we now give detail on the different types of symmetry operation that we will test for in our symmetry finding implementation. Perhaps the most interesting type of symmetry is the translational symmetry of the configuration. The translational symmetry can be defined by a pair of generating translation vectors. Two shortest linearly independent translations are always guaranteed to form one such pair [91]. We test all the translation vectors within a double cover of the unit cell, shown in Figure 3.16, to find two such shortest linearly independent translations.

The primitive cell with highest symmetry is a canonical choice and can be found by using the two shortest linearly independent vectors, as shown in Figure 3.17. The two shortest linearly independent vectors form the sides of the canonical primitive cell. There is an exception: if the two shortest linearly independent vectors have unequal length and can be added together to create another linearly independent vector with the same length as the second shortest vector, then that vector along with the other second shortest vector form the sides of the canonical

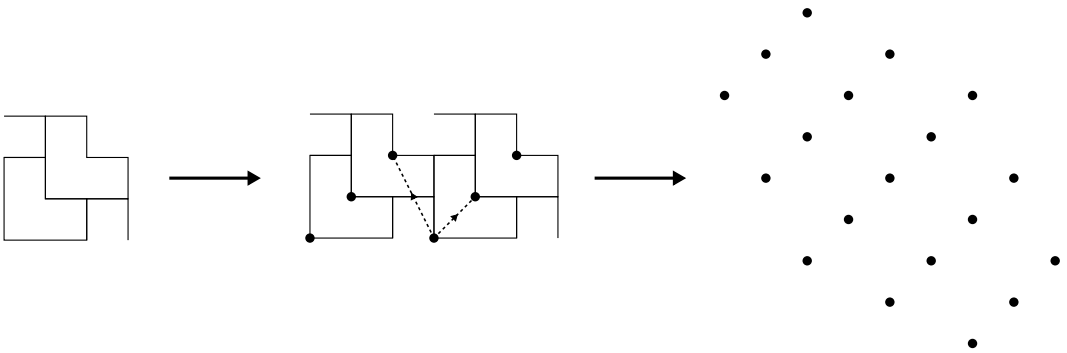


Figure 3.16: A right tromino configuration on the 3×3 unit cell is shown. Two linearly independent shortest translation vectors can be found within a double cover. These translation vectors generate the translational lattice of the polyomino pattern.

primitive cell. This case is labelled as Rhombic 1, in Figure 3.17. The concept of describing the lattice by using shortest length translation vectors is described in Ref. [86].

The shape of the canonical primitive cell also gives a natural categorisation for the lattice, called the lattice type. Excluding the hexagonal lattice, there are 5 lattice types, which are labelled in Figure 3.17. If the shape of the canonical primitive cell is square, rectangular, or rhombic, then the lattice type is square, rectangular, or rhombic, respectively. If the shape has no symmetry, then the lattice type is oblique. The wallpaper group of a given pattern can only have symmetries that are compatible with the lattice type. For example, if a pattern has oblique lattice type, the wallpaper group cannot be p4 because the translational symmetries are not compatible with 4-fold rotational symmetry. This concept plays an important part in the structure of the flow diagrams described in Subsection 3.3.3.

Symmetry operations that preserve the polyomino configuration on the unit cell are not guaranteed to preserve the corresponding planar pattern. However, if a symmetry operation preserves the configuration on the unit cell and maps the contents of the canonical primitive cell to an equivalent canonical primitive cell at another location, the symmetry preserves the corresponding planar pattern. This is because the planar pattern can be thought of as being tiled by the canonical primitive cell. Therefore, the symmetry check on one canonical primitive cell holds true for all the other canonical primitive cells. Since they tile the plane, the symmetry check is sufficient for the entire plane. The important point here is that in our symmetry detection algorithm, attention must be paid to only test for symmetry operations on the unit cell that preserve the canonical primitive cell. This concept is illustrated in Figure 3.18.

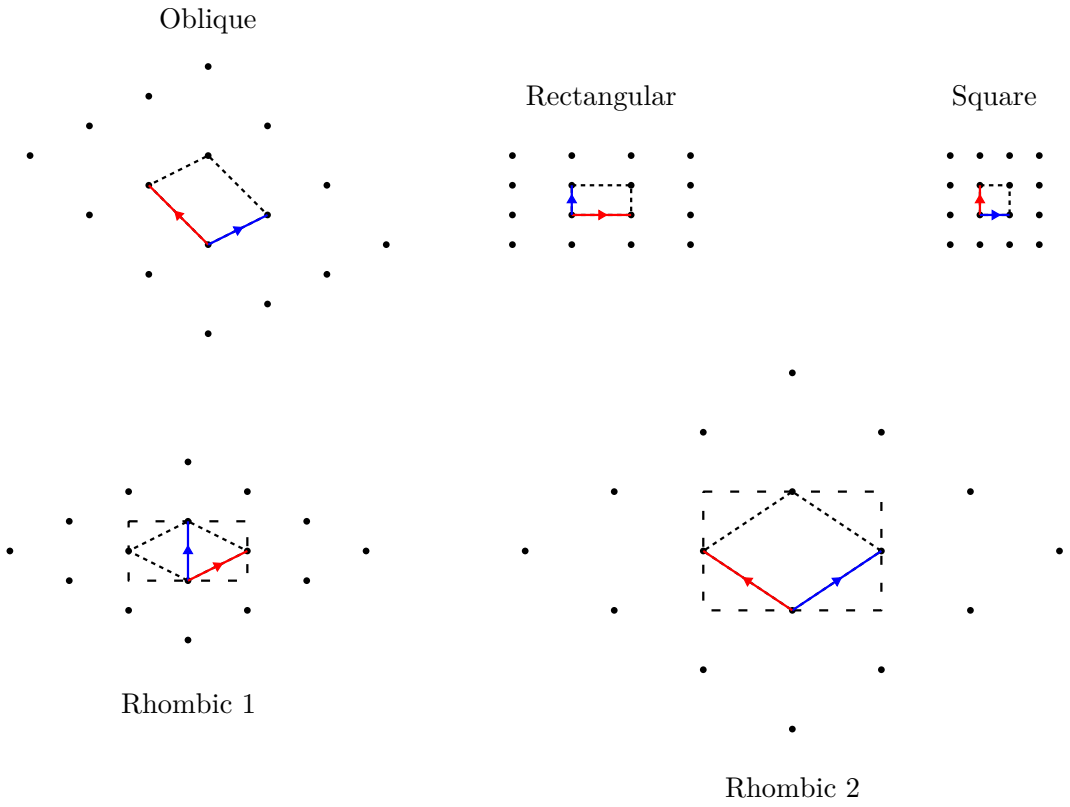


Figure 3.17: Lattices are shown, along with dashed lines representing their canonical primitive cells. Blue arrows correspond to shortest vectors and red arrows correspond to second shortest linearly independent vectors. Not including the hexagonal, there are 4 lattice types. However, the Rhombic type can be further subdivided into two types, depending on how its shortest vectors relate to its canonical primitive cell. Also, for the rhombic type, the centred cell is given by dashed lines with wider spacing between dashes.

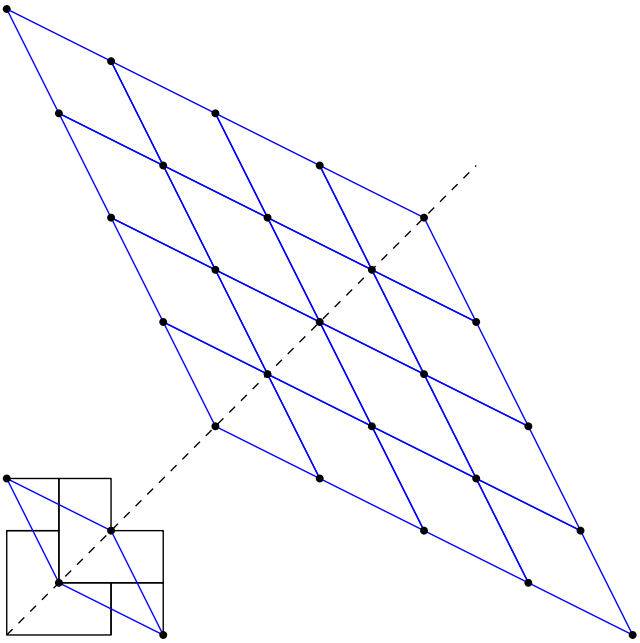


Figure 3.18: Right tromino configuration on the 3×3 unit cell, with a mirror symmetry shown as a dashed line. The canonical primitive cell is shown in blue. The contents of the canonical primitive cell are preserved by the symmetry operation, and looking at the tiling of the plane by canonical primitive cells in the upper-right of the Figure, this symmetry operation will preserve the corresponding planar pattern.

The transformations we must consider are those given in Equation (3.7), i.e. the symmetry operations of the underlying grid, but with the translation vectors being restricted to within the unit cell, since those outside are superfluous. When this set of transformations is subdivided into pure translations, rotations, mirrors, and glides, the subset of transformations that are possible for polyomino configurations becomes clearer. For example, mirror symmetries are only possible along axes parallel or diagonal to the underlying grid.

The image labelled p4m in Figure 2.7 shows the wallpaper group of the square grid, with symmetries visually depicted within the canonical primitive cell. This image gives the symmetries that would be seen within a single square of the square grid. The list of various symmetry types with the parameter values that preserve the underlying square grid is given in Figure 3.19. When testing for symmetry operations, we will only be using the operations listed in Figure 3.19, along with pure translations, to make use of the fact that some symmetries can be disregarded due to incompatibility with the underlying square grid.

3.3.3 Flow Diagrams For Symmetry-Finding

Having a set of possible symmetries and a means to test them, we can now implement an algorithm to find the wallpaper group of the polyomino pattern. A naïve algorithm would test all possible symmetries, store them in a list, then analyse that list to find the wallpaper group. However, this would be quite wasteful, as the existence of some symmetries can imply or disqualify others. Therefore, we implement a strategy which makes use of the information that has already been found by the algorithm to narrow down the possibilities until the wallpaper group is found.

Firstly, we find the two shortest linearly independent translation vectors, establish the canonical primitive cell, and identify the lattice type using the method described in Subsection 3.3.2. The translational symmetry is an important part of the rest of the algorithm, so we use T_1 and T_2 to denote the translation vectors of the sides of the canonical primitive cell, except in the rhombic lattice, where they denote the sides of the centred unit cell.

As can be seen in Figure 3.19, the mirror or glide reflections must be parallel or diagonal to the underlying square grid. However, they must also be parallel or diagonal to T_1 and T_2, as shown in Figure 2.7. Therefore, if T_1 and T_2 are not parallel or diagonal to the underlying square grid, we can immediately deduce that there can be no reflection or glide transformations. In this special case, we simply need to test for 2-fold and 4-fold rotations, which gives either p1, p2, or p4 as the wallpaper group.

$$\begin{pmatrix} x \\ y \end{pmatrix} \to \begin{pmatrix} 0 & -1 \\ 1 & 0 \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} + \begin{pmatrix} z \\ w \end{pmatrix}$$

$\pi/2$ Rotation around $\left(\frac{z-w}{2}, \frac{z+w}{2}\right)$

$$\begin{pmatrix} x \\ y \end{pmatrix} \to \begin{pmatrix} -1 & 0 \\ 0 & -1 \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} + \begin{pmatrix} z \\ w \end{pmatrix}$$

π Rotation around $\left(\frac{z}{2}, \frac{w}{2}\right)$

$$\begin{pmatrix} x \\ y \end{pmatrix} \to \begin{pmatrix} 0 & 1 \\ -1 & 0 \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} + \begin{pmatrix} z \\ w \end{pmatrix}$$

$3\pi/2$ Rotation around $\left(\frac{w+z}{2}, \frac{w-z}{2}\right)$

$$\begin{pmatrix} x \\ y \end{pmatrix} \to \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} + \begin{pmatrix} z \\ w \end{pmatrix}$$

Glide through axis $y = x + \frac{w-z}{2}$
with translation $\frac{w+z}{2}(1,1)$

$$\begin{pmatrix} x \\ y \end{pmatrix} \to \begin{pmatrix} 0 & -1 \\ -1 & 0 \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} + \begin{pmatrix} z \\ w \end{pmatrix}$$

Glide through axis $y = -x + \frac{z+w}{2}$
with translation $\frac{z-w}{2}(1,-1)$

$$\begin{pmatrix} x \\ y \end{pmatrix} \to \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} + \begin{pmatrix} z \\ w \end{pmatrix}$$

Glide through axis $y = \frac{w}{2}$
with translation $(z, 0)$

$$\begin{pmatrix} x \\ y \end{pmatrix} \to \begin{pmatrix} -1 & 0 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} + \begin{pmatrix} z \\ w \end{pmatrix}$$

Glide through axis $x = \frac{z}{2}$
with translation $(0, w)$

Figure 3.19: The different types of symmetry transformations that preserve the square grid are listed, except the pure translations. The values z and w are integers. Mirror symmetries are glides with zero translational component.

Except in the preceding special case, our algorithm splits for each of the lattice types. The lattice type can immediately disqualify several wallpaper groups, and there are elements in common between wallpaper groups of the same lattice type. For example, it is only necessary to check for 4-fold rotational symmetry when the lattice type is square. For this reason, it is natural to split the algorithm into routines for each lattice type.

The algorithms to determine the wallpaper group, starting from the lattice type, are given in the form of flow diagrams in Figure 3.20 and Figure 3.21. In the flow diagrams, the possible remaining wallpaper groups are shown at each step along with the next symmetry to be tested, which results in a yes or no answer. The result of the test determines the branch that the algorithm will continue along. Various tests are repeated until the wallpaper group is identified. Since some of the wallpaper groups are subgroups of others and positive results of symmetry tests continue towards the lower-right, each wallpaper group must appear to the right of its subgroups. For example, p4 is a subgroup of the p4m wallpaper group; therefore, p4 must be to the left of p4m. In this sense, the high symmetry wallpaper groups are to the right of the flow diagram.

At each step of the flow diagrams, we test a family of symmetry operations. For example, testing 4-fold rotations involves testing a rotation of $\pi/2$ at each position of the unit cell that is compatible with the underlying grid (i.e. of the form given in Figure 3.19). If any one of these symmetry operations preserves the polyomino configuration, the algorithm stops the search and continues along the yes path; otherwise, it continues along the no path. For a given vector v, testing for v mirror means testing for all possible mirrors parallel to v. Testing v glide means to test all glides parallel to v with translation component equal to half the magnitude of v. Also note that $T_1 \pm T_2$ are vectors diagonal to T_1, T_2. These diagonal vectors are used in the square lattice algorithm for two reasons: Firstly, to test for the mirrors of p4g, which are diagonal to T_1, T_2. Secondly, to test for the mirrors of cm and cmm, which are diagonal to T_1, T_2 for the square lattice, but not for the rhombic lattice, since a centred cell is used in that case.

The design of our symmetry-finding algorithm is determined by the possible ways that symmetries can combine to form the wallpaper groups; an efficient flow diagram disqualifies wallpaper groups with as few symmetry tests as possible. The illustrations of Figure 2.7 show how the symmetries combine to give the wallpaper groups. Here, we give a short explanation to accompany the visual flow diagrams of Figure 3.20 and Figure 3.21. If the lattice type is oblique, the wallpaper group must be either P1 or P2. This is the easiest case because we only need to test for 2-fold

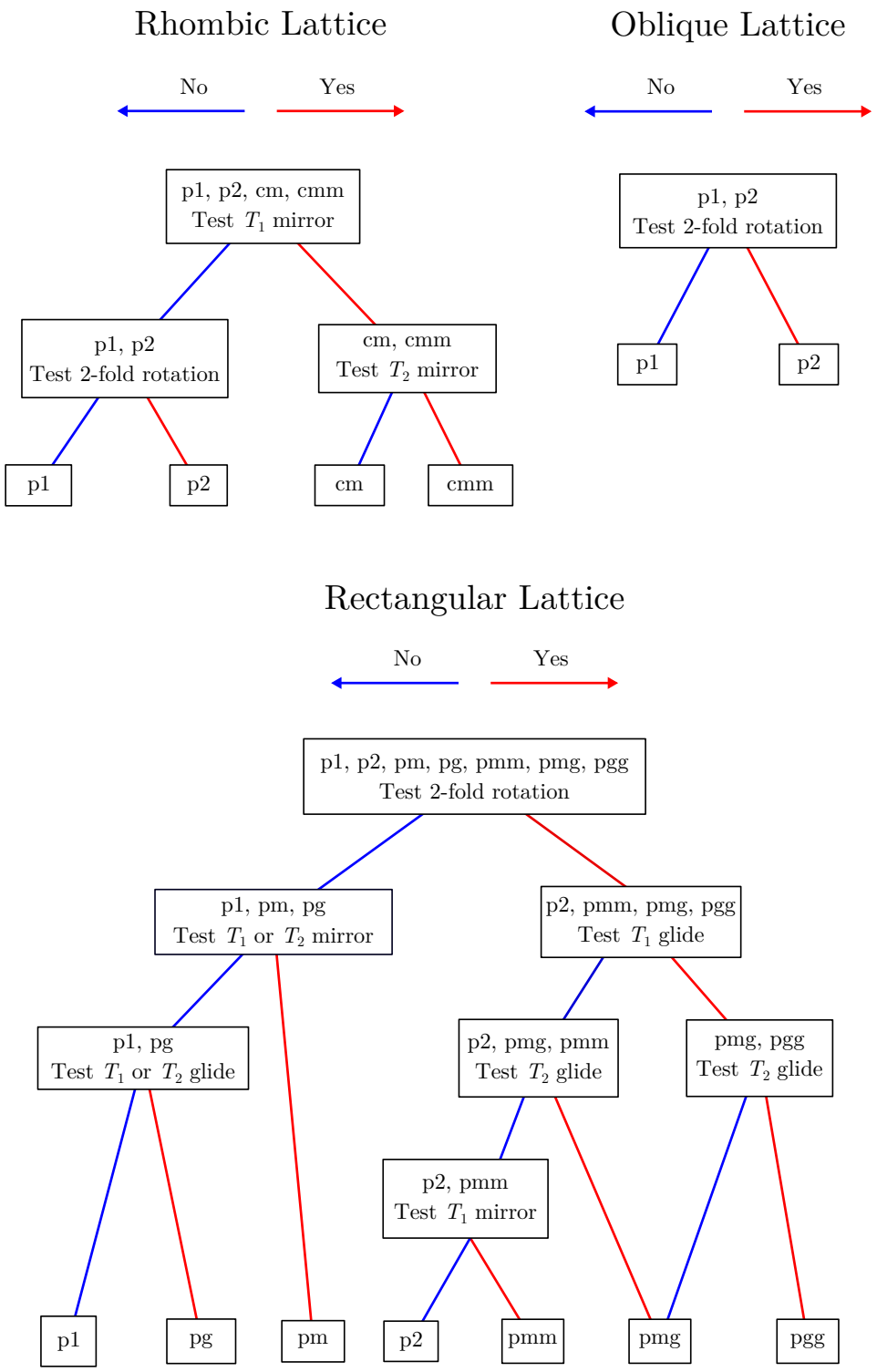


Figure 3.20: Flow diagrams for symmetry detection in rhombic, oblique, and rectangular lattice types. Each diagram progresses downwards, along the branch indicated by the test in each box. The possible wallpaper groups remaining are also shown at each stage of the diagram.

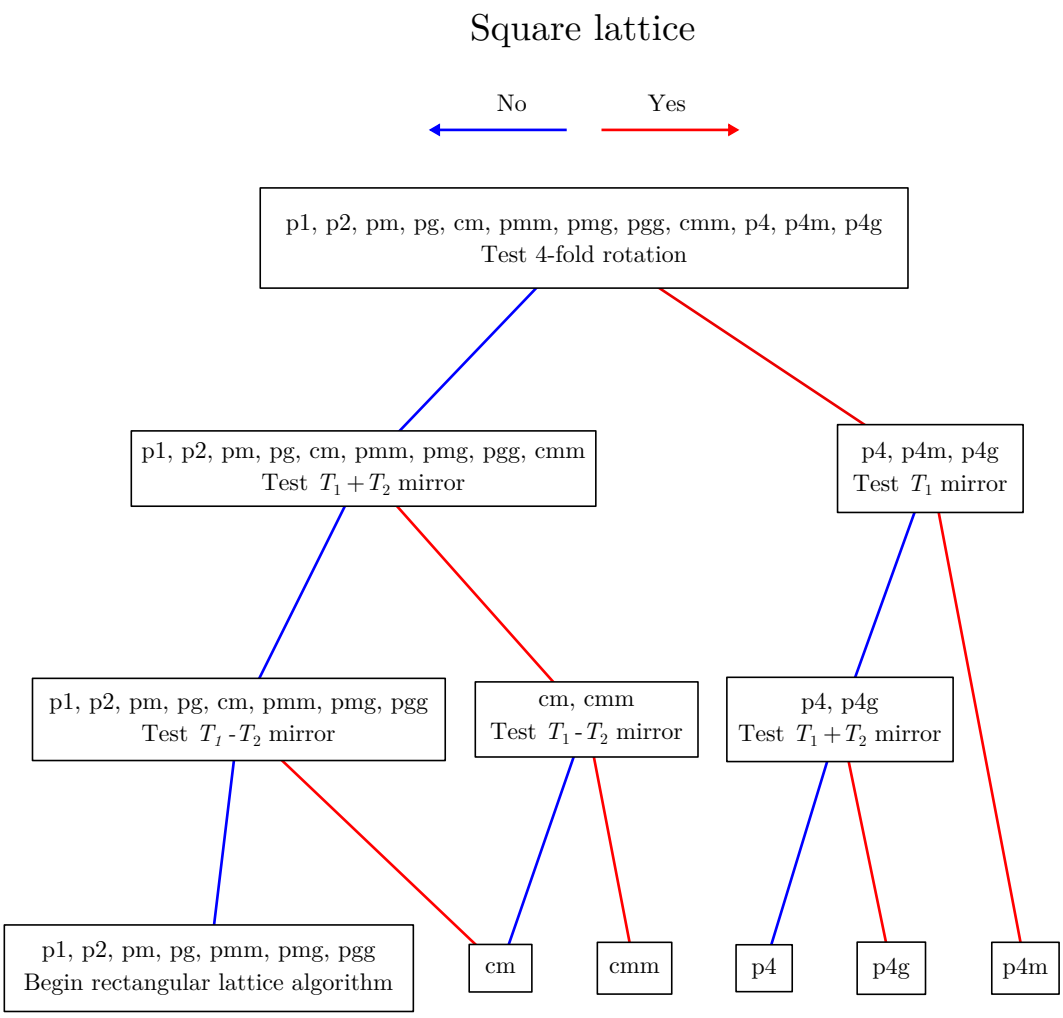


Figure 3.21: Flow diagram for symmetry detection in the square lattice scenario. The diagram progresses downwards, along the branch indicated by the test in each box. The possible wallpaper groups remaining are also shown at each stage of the diagram. One of the final stages results in commencing the rectangular lattice algorithm shown in the flow diagram of Figure 3.20 in order to identify the wallpaper group.

rotations to distinguish the wallpaper group. Otherwise, if the lattice is rhombic, we have an unusual case because we need to check for mirrors parallel to the centred cell translation vectors, not the canonical primitive cell. The rhombic lattice can result in either of: Cmm, Cm, P2, or P1, depending on which mirrors or rotations are present.

For the rectangular lattice, we check for 2-fold rotations, mirrors, and glides (which can be in either of the directions T_1 or T_2). This will result in one of the wallpaper groups P1, P2, Pm, Pg, Pgg, Pmg, or Pmm. Finally, we have the square lattice case. This is the most difficult, since the wallpaper group can be any one of the possible 12. For the square lattice, we first test if there are 4-fold symmetries, and in that case we test mirror symmetries to further classify as P4, P4g, or P4m. If there are no 4-fold rotations, we check if the wallpaper group is Cm or Cmm by testing diagonal mirrors. If the wallpaper group is neither of these, we continue by doing all of the tests that are used for the rectangular lattice.

The algorithm described above is the method we have used to find the wallpaper group and primitive cell of polyomino configurations. Specifically, it takes a polyomino configuration on a unit cell and calculates the wallpaper group that would result from tiling that cell across the plane. As mentioned, the two main benefits of our algorithm are that we take advantage of the fact that we only need to consider the symmetries of the underlying square grid, and the structure of the algorithm allows us to classify the wallpaper group without needing to check all the possible symmetry transformations. In fact, knowledge of the wallpaper group and primitive cell can be used to reproduce all the symmetries of the pattern. In this sense, our algorithm implicitly finds all symmetries of the pattern. Figure 3.22 gives an example of the algorithm working on a domino configuration.

3.4 Sublattices

3.4.1 Fast Check For No Extra Symmetries

In the previous section, we described an algorithm to determine the symmetries of a polyomino configuration. When analysing many such configurations, it becomes important to try to further improve on the efficiency. This can be done by looking for a bottleneck in the algorithm; in other words, the part of the algorithm that uses most of the computational resource. In our case, the algorithm becomes slow when the unit cell is large and many polyomino configurations are being enumerated. For large unit cells, the vast majority of polyomino patterns have zero symmetry, apart from the unit cell translations, which are guaranteed. As an example, when enu-

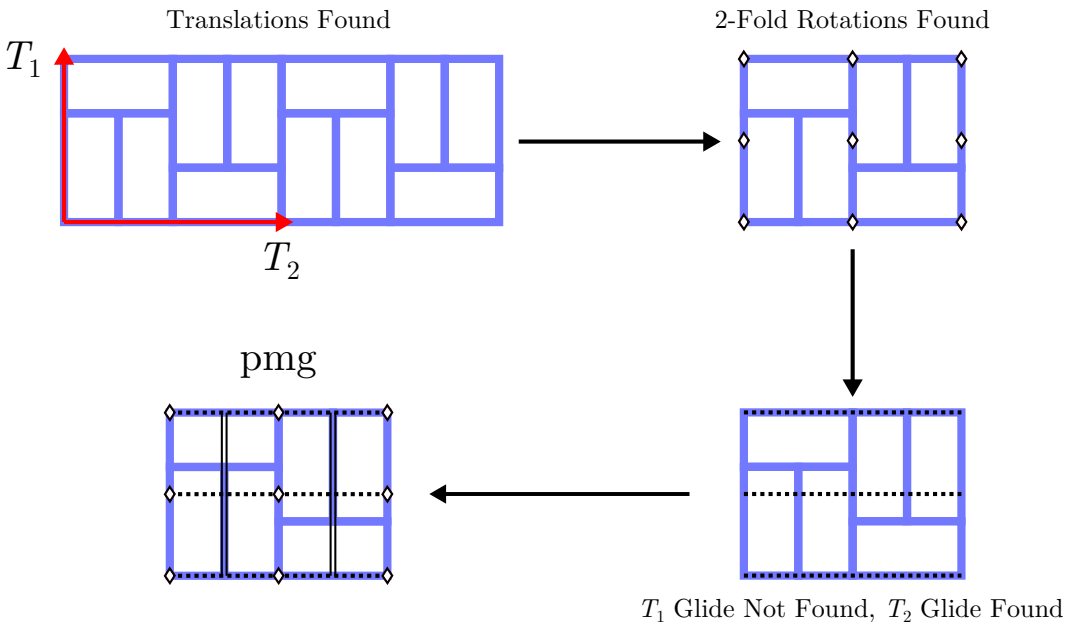


Figure 3.22: A domino configuration with pmg wallpaper group on 8×3 unit cell is analysed by the symmetry-finding algorithm. The diamonds signify 2-fold rotation centres, the dashed lines signify glide reflections, and the double lines signify mirrors. Once a single 2-fold symmetry is found, all the rest are implied by the translational symmetry. Similarly, the combination of T_2 glide and rotations implies the T_1 mirror, which does not need to be tested explicitly.

merating domino patterns on an 8×8 unit cell, 1,181,212 out of 1,224,518 (roughly 96%) of the polyomino patterns have no extra symmetries. This is a bottleneck of our algorithm, since most of the computational resource is used in confirming that these patterns have no extra symmetries.

To address this bottleneck, we have made an addition to our symmetry algorithm, which first makes an efficient check to determine whether there are any extra symmetries at all. Then, if there is at least one symmetry, we continue to analyse the other possible symmetries as described in the previous section. A major component of this fast preliminary algorithm is the fast check to confirm there are no extra translational symmetries; it is also necessary to make a fast check for symmetries other than translations. Individual translational symmetries are tested in the same way as in the main algorithm of the previous section. The difference is that in the fast check for translations, we only test the theoretical minimum number of translational symmetries that are required to be able to decide whether there are any extra translations or not.

Our fast check for translations makes use of a few group theory concepts. The full group of possible translational symmetries is $\mathbb{Z} \times \mathbb{Z}$. Since this is an Abelian group, all subgroups are normal and we are able to take a quotient using the translations of the unit cell $(\mathbb{Z} \times \mathbb{Z})/(N\mathbb{Z} \times M\mathbb{Z})$, giving the group $\mathbb{Z}_N \times \mathbb{Z}_M$ as our group of possible symmetries. This group represents the possible translational symmetries that are in addition to the translations of the original unit cell. To be specific, the elements of $\mathbb{Z}_N \times \mathbb{Z}_M$ can be written as (x, y) with $0 \leq x < N$ and $0 \leq y < M$ and group operation $(x, y) + (z, w) = ((x + z) \bmod(N), (y + w) \bmod(M))$.

The translational symmetries of our polyomino pattern must be some subgroup of $\mathbb{Z}_N \times \mathbb{Z}_M$. The trivial subgroup corresponds to a polyomino pattern that only has the translations of the original unit cell. In this fast algorithm, the aim is to use as few symmetry tests as possible to confirm that the subgroup of pattern-preserving symmetries is the trivial group.

When a test for translational symmetry returns a negative result, we say the symmetry is disqualified. We can also think in terms of disqualifying subgroups. The subgroup of $\mathbb{Z}_N \times \mathbb{Z}_M$ that corresponds to the symmetries of the pattern must contain only elements that preserve the pattern. Therefore, if a subgroup contains a disqualified element, that subgroup is disqualified. Furthermore, each subgroup containing that subgroup is also disqualified for the same reason. Thinking in this way, the subgroups form a lattice by inclusion, with the smallest subgroup being the trivial group. Disqualifying one subgroup will disqualify all subgroups that are greater than it in the lattice order. The lattice of subgroups for $\mathbb{Z}_4 \times \mathbb{Z}_2$ is shown in

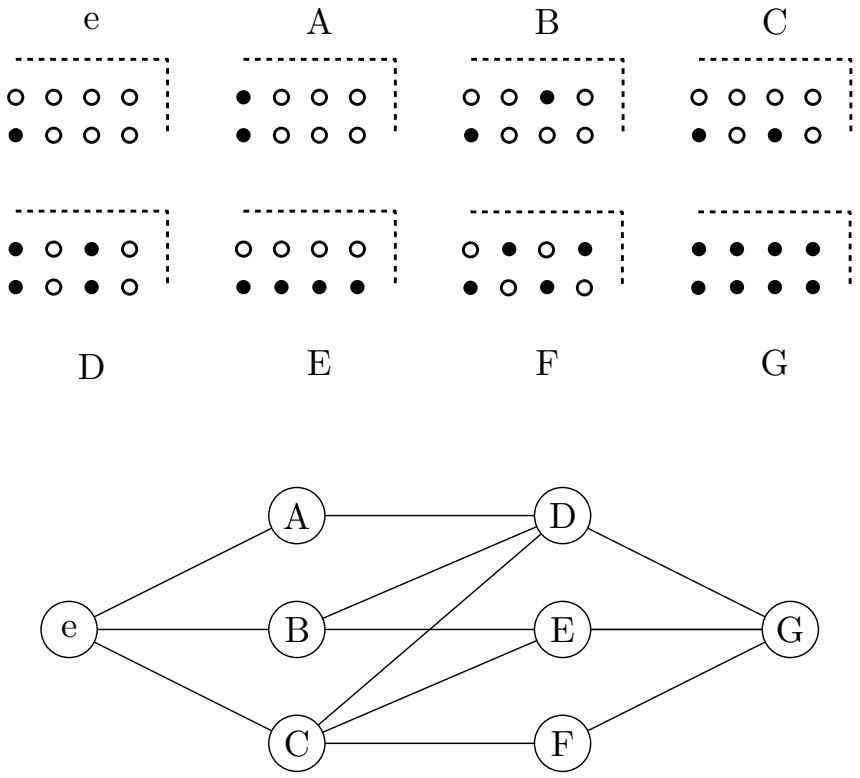


Figure 3.23: Top: all subgroups of $\mathbb{Z}_4 \times \mathbb{Z}_2$ are shown by using filled dots to indicate positions that are related to the origin by the translations defining the subgroup. The dashed line can be thought of as the upper-right side of the periodic boundary. Below: The corresponding lattice of subgroups, using order by inclusion. Lines between two subgroups indicate that the subgroup on the left is covered by the subgroup on the right.

the lower part of Figure 3.23. For example, the subgroups greater than subgroup B are the subgroups D, E, and G.

Keeping the lattice of subgroups in mind, we must explicitly disqualify at least one element from each of the subgroups that has no nontrivial subgroups of its own, since these are the smallest nontrivial subgroups in the lattice order. Furthermore, all of the other subgroups are greater than at least one of these smallest subgroups. Therefore, by disqualifying all the smallest subgroups in the lattice order, the subgroup of symmetries can be confirmed as the trivial group. In Figure 3.23, the subgroups we need to explicitly disqualify are the subgroups A, B, and C. A well-known group theory result is that groups with no nontrivial subgroups are exactly the cyclic groups of prime order. Therefore, it is the prime cyclic subgroups that we must disqualify. Without them, only the trivial subgroup is possible. We will consider the trivial group to not be prime cyclic, to aid the simplicity of discussion.

Before being able to disqualify the prime cyclic subgroups, we must identify them. Suppose an element (x, y) of $\mathbb{Z}_N \times \mathbb{Z}_M$ generates a prime cyclic subgroup with order p, which divides N, M, or both. We can investigate the possible cases. Suppose p divides N but not M. Since the element generates a prime cyclic subgroup, $py = \beta M$ for some integer β. Combining this with the facts that $y < M$ and p is a prime that doesn't divide M, we must have $y = 0$. Furthermore, $px = \alpha N$ for some integer α, and since p divides N, we can simply write $x = \alpha(N/p)$ to identify all elements that generate prime cyclic subgroups of order p. Since $0 \leq x < N$, the elements of the subgroup are $(\alpha(N/p), 0)$ with $0 \leq \alpha < (N/p)$. These elements form a single prime cyclic subgroup of horizontal translations. If instead we have p that divides M but not N, there is an analogous situation, giving a prime cyclic subgroup of vertical translations. An example of this is shown in Figure 3.24, for $p = 2$, which divides $M = 6$, but not $N = 9$.

The more complicated case occurs when p divides both N and M. For this case, we can define the set of elements that generate prime cyclic subgroups of order p by a reasoning similar to that used in the previous case, giving elements $(\alpha(N/p), \beta(M/p))$ with $0 \leq \alpha < (N/p)$ and $0 \leq \beta < (M/p)$. However, this is not a prime cyclic subgroup, but is isomorphic to the group $\mathbb{Z}_p^2$. This group contains p^2 elements, each being part of a prime cyclic subgroup of p elements that don't overlap, apart from the identify. Therefore, there are $p + 1$ prime cyclic subgroups of order p. Figure 3.24 gives an example, where $p = 3$ divides both $M = 6$ and $N = 9$. Also, the 4 prime cyclic subgroups are indicated.

In conclusion, for a given unit cell of dimensions $N \times M$, it is necessary to list all the primes that divide N or M. For a given prime p, if it divides N but not M, there is only one prime cyclic subgroup of that order, which can be tested by element $(N/p, 0)$. An analogous case holds for a prime that divides M but not N. For a prime p that divides both N and M, there are $p + 1$ prime cyclic subgroups of that order, which can be tested by elements $(N/p, 0)$ and $(\alpha(N/p), M/p)$ for $0 \leq \alpha < (N/p)$. Using Figure 3.24 as an example, the primes that divide 9 or 6 are 2 and 3. The prime 2 divides only 6, so it can be tested by using element $(3, 0)$. The prime 3 divides both 9 and 6, so there are 4 prime cyclic subgroups, which can be tested by elements $(3, 0)$, $(0, 2)$, $(3, 2)$, and $(6, 2)$. This compares well to the straightforward implementation, which would need to test all 53 nontrivial elements.

This method gives a fast check to confirm no extra translational symmetries. Indeed, it is the theoretical minimum number of checks required, since each prime cyclic subgroup must be disqualified, none of which overlap with each other, and we only test one translation for each prime cyclic subgroup. To finish the fast

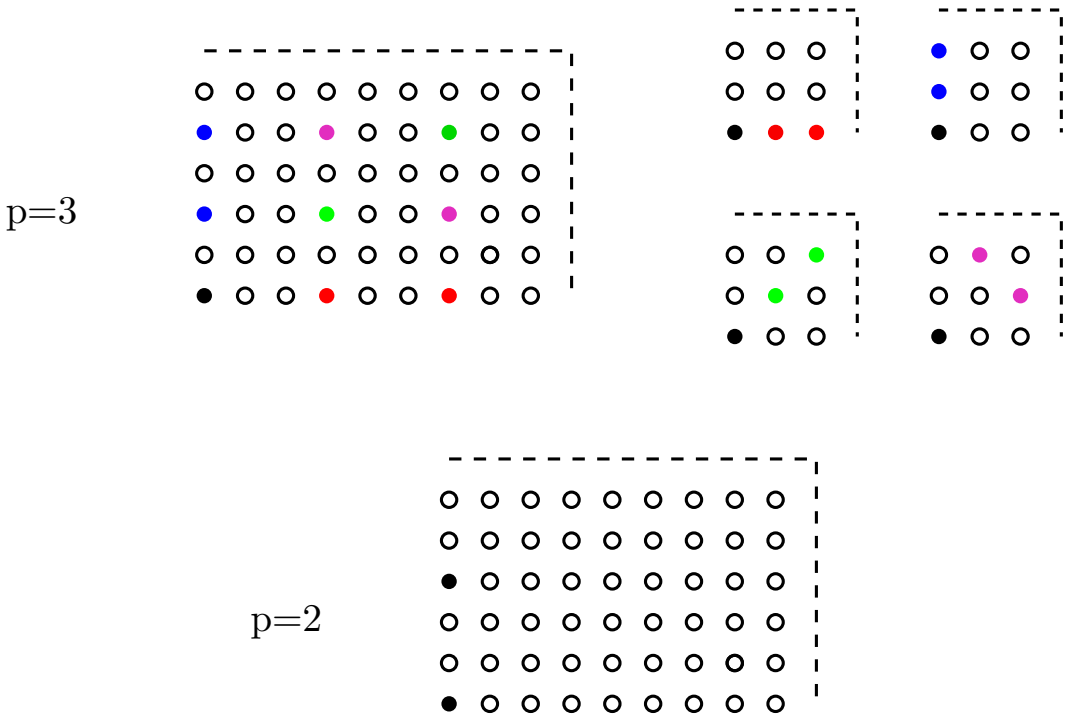


Figure 3.24: The prime cyclic subgroups are shown for $\mathbb{Z}_9 \times \mathbb{Z}_6$. Both sides are divisible by 3, so the elements of order 3 form a subgroup $\mathbb{Z}_3^2$, shown in the top-left. Within this subgroup are 4 prime cyclic subgroups, shown top-right. Colours correspond to each of the prime cyclic subgroups, except the identity element, which is coloured black. Only the vertical side of $\mathbb{Z}_9 \times \mathbb{Z}_6$ is divisible by 2, so there is only one prime cyclic subgroup of order 2, shown at the bottom of the figure. The only divisors of either side are 2 and 3; therefore, there are no other prime cyclic subgroups. In total, $\mathbb{Z}_9 \times \mathbb{Z}_6$ has 5 prime cyclic subgroups.

check for no extra symmetries, we must test for other types of symmetry apart from translation. We must test for 2-fold rotation, T_1 and T_2 mirrors and glides, $T_1 + T_2$ glides, and $T_1 - T_2$ glides. These tests are necessary to remove the possibility of the pattern having extra symmetries that are non-translational.

If a symmetry is detected at any point in the fast algorithm, the main symmetry-finding algorithm from the previous section begins. Otherwise, the wallpaper group is simply p1, and the translations of the polyomino pattern are those of the unit cell. This method makes it possible to analyse the wallpaper groups of a large number of patterns. Interestingly, the fast check for translations makes the best savings on computational resource when the unit cell sides are powers of a single prime. For the example of a square cell with sides a power of 2, only 3 translations need to be tested: $(N/2, 0)$, $(0, M/2)$, and $(N/2, M/2)$.

3.4.2 Rectangular Sublattices

Our method for enumerating planar polyomino patterns involves generating configurations on a rectangular unit cell and picturing this unit cell as tiling the plane to create a planar pattern. Thinking in the reverse direction, what can we say about the rectangular unit cells that reproduce a given planar polyomino pattern? In terms of the translational lattice of a periodic planar polyomino pattern, the rectangular unit cells that generate it can be thought of as rectangular sublattices that have shortest linearly independent translations that are parallel to the underlying grid. In this subsection, we give details on the set of such sublattices for a general periodic planar polyomino pattern. In doing this, we can understand the kind of unit cells required to generate particular patterns.

For any given polyomino pattern, we can choose its generating translations to be in Hermite normal form: (γ, α) and $(0, \beta)$. As shown in Equation (3.3) of Subsection 3.2.2, any rectangular sublattice must satisfy the conditions $\gamma\beta \mid N \gcd(\alpha, \beta)$ and $\beta \mid M$, where $(N, 0)$ and $(0, M)$ are the generating translation vectors of the rectangular sublattice. Therefore, the smallest rectangular unit cell that generates the pattern has cell sides $(\gamma\beta / \gcd(\alpha, \beta)) \times \beta$. Furthermore, any other rectangle that generates the pattern is formed by a collection of copies of the smallest rectangle.

The smallest square sublattice of a given pattern can be found by calculating the smallest square that is commensurate with the rectangle $(\gamma\beta / \gcd(\alpha, \beta)) \times \beta$, and larger than it. In other words, we must find the smallest value L for which there exists integers a,d that satisfy the equation

$$\begin{pmatrix} L & 0 \\ 0 & L \end{pmatrix} = \begin{pmatrix} a & 0 \\ 0 & d \end{pmatrix} \begin{pmatrix} \gamma \frac{\beta}{\gcd(\alpha,\beta)} & 0 \\ 0 & \beta \end{pmatrix}. \tag{3.8}$$

To satisfy this, we need $a\gamma = d\gcd(\alpha, \beta)$, resulting in the smallest value of d being

$$d = \frac{\gamma}{\gcd(\gamma, \gcd(\alpha, \beta))}, \tag{3.9}$$

and multiplying by β to obtain L, we have the value

$$L = \frac{\beta\gamma}{\gcd(\gamma, \alpha, \beta)}. \tag{3.10}$$

This gives a derivation for the result that was claimed in Subsection 3.2.2.

We can also consider what are the most awkward polyomino patterns to generate by using a rectangular unit cell. In other words, the patterns with primitive unit cell area being the smallest fraction of the area of the smallest rectangular unit cell that can generate the pattern. As mentioned, the smallest rectangular unit cell that generates a pattern with Hermite normal form translations (γ, α) and $(0, \beta)$ is $(\gamma\beta/\gcd(\alpha, \beta)) \times \beta$. Consequently, the most awkward translational lattices have parameters $\gamma = 1$ and $\alpha = 1$. In this case, the area of the primitive cell is β and the smallest rectangular unit cell that generates the pattern is the square with sides β. An example is shown in Figure 3.25 for $\beta = 3$.

We have shown that given a general polyomino pattern with area of primitive cell A, a square cell with sides of length A is necessary to guarantee it can be generated, since it may have an awkward lattice of translations. We also consider the converse, does a square cell with sides of length A generate all polyomino patterns with primitive cell area A? To establish that this is true, we use the result from Equation (3.10). This equation shows that the smallest square sublattice of a general polyomino pattern has sides of length that divide $\gamma\beta$. The value $\gamma\beta$ is simply the area of the primitive cell; therefore, each polyomino pattern can be generated by a square unit cell of sides equal to the area of its primitive unit cell $\gamma\beta$. In other words, the square unit cell of sides A generates all polyomino patterns of primitive unit cell area A.

Using the two previous points, the square unit cell with sides A is necessary and sufficient to generate all polyomino patterns with primitive unit cell area A. Furthermore, by using square unit cells with area A from $A_{\max}/2$ to $A_{\max}$, it is possible to generate all polyomino patterns with primitive unit cell area up to $A_{\max}$. The square unit cells of area below $A_{\max}/2$ are not necessary because they are contained in the larger unit cells. This gives a precise sense in which it is possible

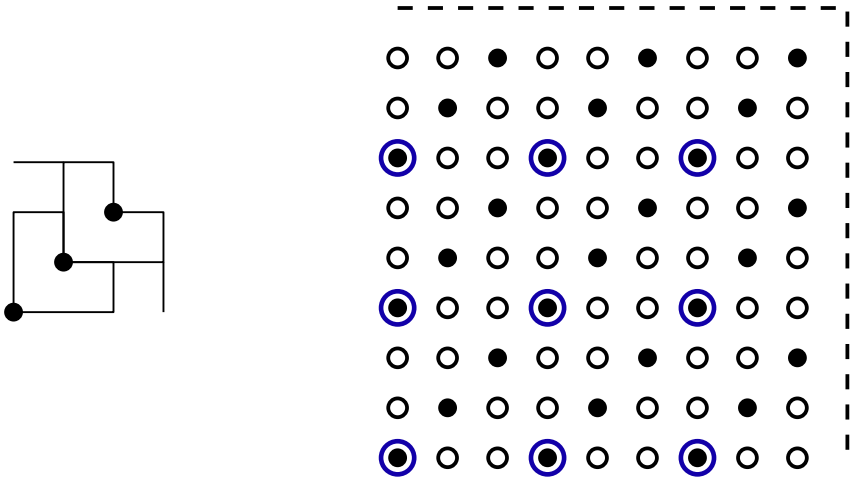


Figure 3.25: A polyomino configuration with an awkward translational lattice. The area of the primitive cell is 3, and the smallest rectangular sublattice is generated by translations $(3, 0)$ and $(0, 3)$. The polyomino configuration is shown within this smallest cell on the left. On the right, the translational lattice of the polyomino configuration is shown by black dots, and the smallest rectangular sublattice is shown as blue circles.

to enumerate the polyomino patterns of low periodicity. For systems that are not effectively discrete, this is not generally possible because there is a continuum of different translational lattices.

3.5 Interactions And Packing Constraints

3.5.1 Model For Interactions

We incorporate simple interactions into our model to represent directional bonds between neighbouring molecules. As explained in Subsection 2.1.1, we define interactions to be between faces of polyominoes that are directly in contact with each other. The interaction strength between two polyomino faces depends on the types of faces involved, and the energy of the polyomino configuration is determined by the interactions that are present in the configuration.

In complete generality, each face of a polyomino will be of a different type. However, we are considering the rotated polyomino to be essentially the same. Without introducing several different polyomino types of the same shape, faces must respect the rotational symmetry of the polyomino. In some situations, it may be possible to introduce a further simplifying set of assumptions; for example, it may be known experimentally that some of the interaction strengths must be zero, or

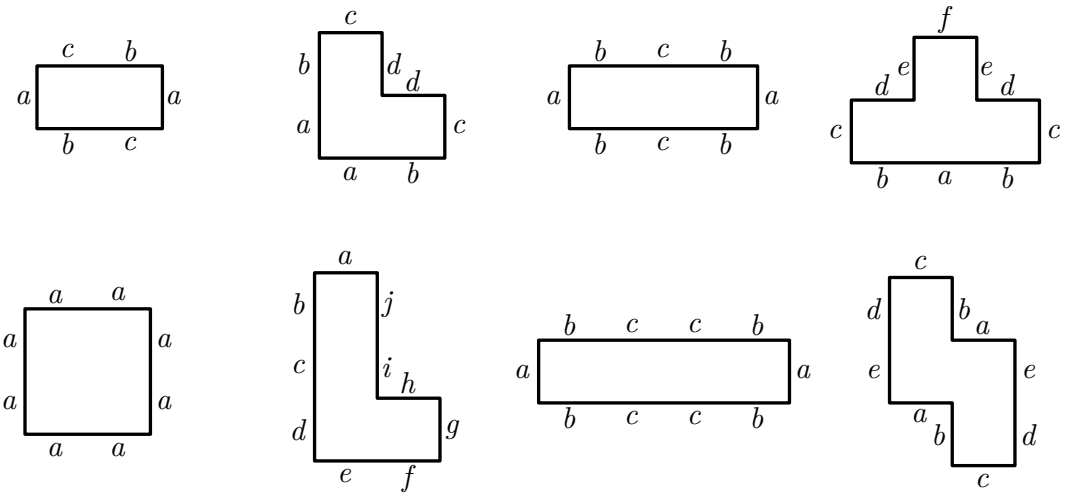


Figure 3.26: The one-sided tetrominoes and trominoes are shown with face types that respect the reflection and rotation symmetry of the polyomino shape. The domino is shown with face types that respect the rotation but not reflection symmetry of the domino. Note that the label conventions are separate for each polyomino. To create a configuration with more than one polyomino type, the labels for faces of different polyomino types would need to be different.

that some are related by symmetry. For the majority of this thesis, the face types of polyominoes will respect both the rotational and reflection symmetries of the polyomino. The main exception is for dominoes, we will not enforce the reflection symmetry of face types so that the variety of interaction parameters are not too restricted. We will use the term chiral for interactions that don't respect the reflection symmetry of the polyomino shape. The polyominoes that we will consider, with their various face types labelled by letters, are shown in Figure 3.26.

Since each face of a polyomino is assigned a type by designating a letter to it, the interaction between two faces can be specified by the two letter combination of the face types in contact. The order of this two letter combination is not important, so by convention we will use the alphabetical order x-y to describe the interaction between faces x and y and refer to it as an interaction type. An example configuration with faces labelled is given in Figure 3.27.

Supposing that we have a configuration with one type of polyomino that has Q face types, there would be $Q(Q+1)/2$ different possible pairs of face types. Each one corresponds to a different interaction type. For each interaction type x-y, we can assign an interaction parameter ε_{xy}. Therefore, there are $Q(Q+1)/2$ different interaction parameters. We can also assign an integer n_{xy} to be the count of the number of interactions of type x-y in the configuration. Furthermore, this

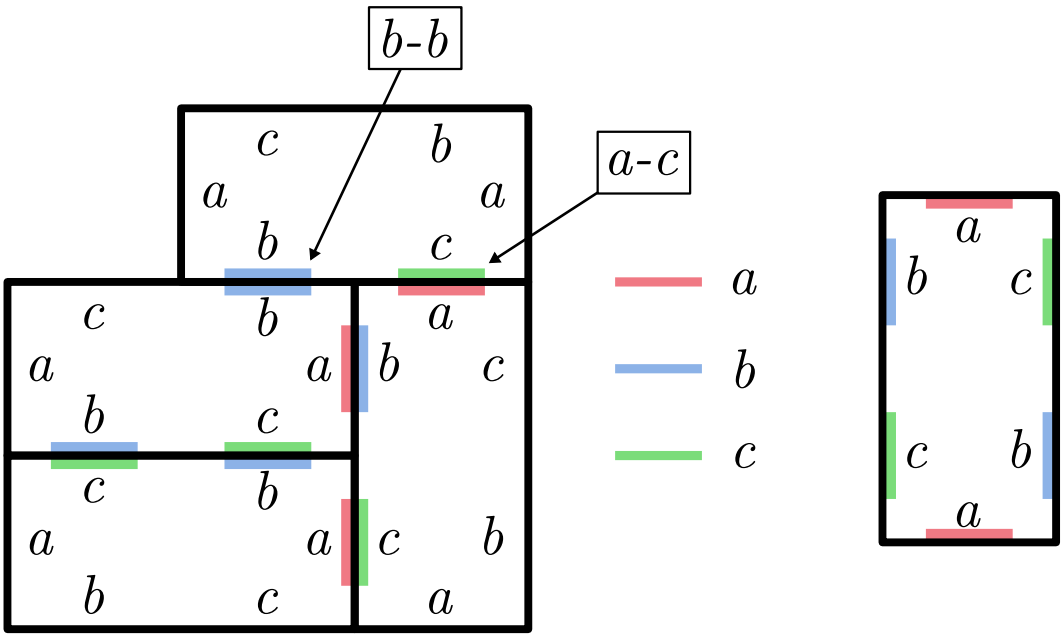


Figure 3.27: Example collection of dominoes with face types labelled and coloured. Each combination of two adjacent face types (or colours) corresponds to a different interaction type. Two of the interaction types are indicated, as b-b and a-c.

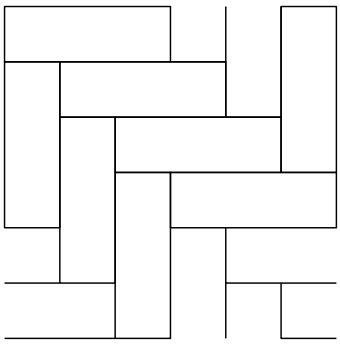
interaction type contributes a value $\varepsilon_{xy} n_{xy}$ to the energy of the configuration. By defining a vector $\vec{n}$ from the elements n_{xy} and a vector $\vec{\varepsilon}$ from the elements ε_{xy}, we see that the overall energy of the polyomino configuration is the dot product $\vec{n} \cdot \vec{\varepsilon}$. We will refer to $\vec{n}$ as the interaction count vector and $\vec{\varepsilon}$ as the interaction parameter vector.

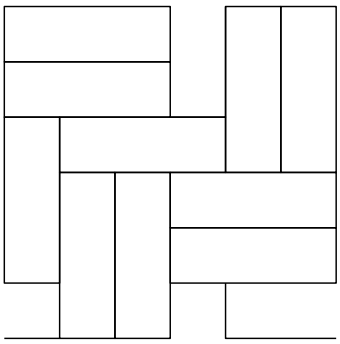
A polyomino system is defined by a set of polyomino configurations and a set of interaction parameters. Once a specific interaction parameter vector $\vec{\varepsilon}$ is chosen, the energy of a configuration in the system with interaction count vector $\vec{n}$ is determined by $\vec{n} \cdot \vec{\varepsilon}$. Therefore, the interaction count vectors of configurations are of central importance in describing the polyomino system. This concept is illustrated in Figure 3.28 for two straight tromino configurations.

Since the energies of configurations in the system depend on the interaction parameters, the properties of the system also depend on the interaction parameters. In this thesis, we will often leave the interaction parameter vector $\vec{\varepsilon}$ as a set of free parameters and explore how the system properties vary with the parameters. To do this, it will be helpful to think of each configuration as existing as a point in an abstract space (call it $\vec{n}$-space), with directions being the counts for each interaction.

n_{aa}	n_{ab}	n_{ac}	n_{bb}	n_{bc}	n_{cc}
0	24	0	0	24	0

n_{aa}	n_{ab}	n_{ac}	n_{bb}	n_{bc}	n_{cc}
0	16	8	12	8	4





Energy: $(24\ \varepsilon_{ab} + 24\ \varepsilon_{bc})$ $(16\ \varepsilon_{ab} + 8\ \varepsilon_{ac} + 12\ \varepsilon_{bb} + 8\ \varepsilon_{bc} + 4\ \varepsilon_{cc})$

Figure 3.28: Two straight tromino configurations on the 6×6 unit cell. Above, their interaction count vectors are given. Below, the energy of each configuration is shown, as dependent on the interaction parameters.

3.5.2 Packing Constraints

For a fully-packed unit cell with one polyomino type, only a subspace of the full $\vec{n}$-space is possible. This is due to packing constraints that reduce the effective dimension of the problem. For a given polyomino, suppose that a specific face type (call it x) appears m_x times on each polyomino and each polyomino takes up s squares of the underlying grid. If we have an $N \times M$ unit cell fully packed with polyominoes of one type, there must be $m_x NM/s$ faces of type x within the unit cell as a whole. Each of these faces of type x can form an x-y interaction with a different face type y or can form an x-x interaction with another face of the same type. An x-x interaction uses up two faces of type x, and an interaction of type x-y uses up one face of type x. Therefore, we obtain the relation

$$\frac{m_x NM}{s} = 2n_{xx} + \sum_{\substack{y \\ y \neq x}} n_{xy}, \tag{3.11}$$

where the left side of the equality is the number of x type faces in the unit cell, and the right side of the equality is calculated by counting up how many faces of type x there must be to form each of the interactions involving x type faces. An analogous equation holds true for each of the face types, giving Q packing constraint equations.

As an example, we can consider straight trominoes on a 6×6 unit cell. In this case, $s = 3$ and the face type b appears $m_b = 4$ times on each straight tromino. Therefore, $48 = 2n_{bb} + n_{ab} + n_{bc}$ gives the packing constraint for face type b. This can be checked on the two example configurations shown in Figure 3.28 of the previous subsection. Furthermore, the face type a appears $m_a = 2$ times on each straight tromino, leading to the equation $24 = 2n_{aa} + n_{ab} + n_{ac}$. Finally, face type c appears $m_c = 2$ times on each straight tromino, leading to the equation $24 = 2n_{cc} + n_{ac} + n_{bc}$.

Using the packing constraint equations, it is possible to create a reduced interaction vector with fewer components than the full interaction count vector, since the other components are specified by the packing constraint equations. This reduced interaction count vector can be chosen in more than one possible way. For simplicity, we will always eliminate the components n_{xx} with same face types. For example, the full interaction count vector for the straight trominoes is given by

$$\vec{n} = (n_{aa}, n_{ab}, n_{ac}, n_{bb}, n_{bc}, n_{cc}), \tag{3.12}$$

but the reduced interaction count vector is (n_{ab}, n_{ac}, n_{bc}). This is shown in Figure 3.29.

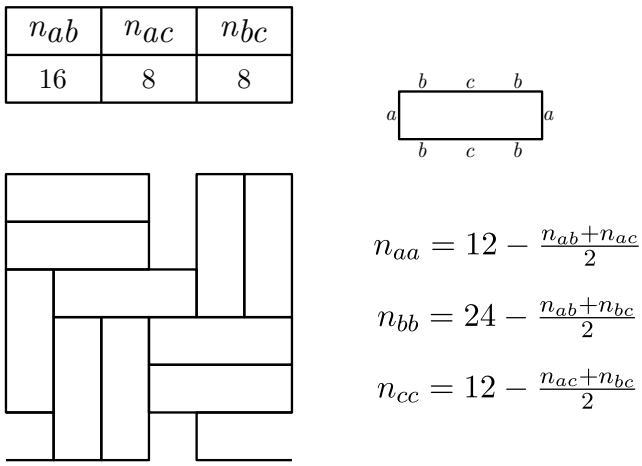


Figure 3.29: For a straight tromino configuration on 6×6 unit cell, the reduced interaction count vector is shown, along with the packing constraint equations, which can be used to obtain the other interaction count values.

In general, the reduced interaction count vector has $Q(Q-1)/2$ components. The reduced interaction count vector implicitly contains all the information about the other interaction counts, so throughout this thesis we will use the reduced interaction count vector, and simply call it the interaction count vector, unless specified otherwise. Also, we redefine the vector of interaction parameters to match the new interaction count vector definition. Using the same example, the energy of a straight tromino configuration is now given by $(n_{ab}\varepsilon_{ab} + n_{ac}\varepsilon_{ac} + n_{bc}\varepsilon_{bc})$. We will continue using the notation $E = \vec{n} \cdot \vec{\varepsilon}$, keeping in mind that $\vec{n}$ no longer corresponds to the full interaction count vector.

3.6 Homology Class Calculation

We have explained in Section 2.3 that the connectivity of domino configurations by domino flip moves can be determined using homology. Pairs of domino configurations for which the oriented transition graph is in the trivial homology class are connected by domino flip moves and can be considered kinetically accessible to each other. To explicitly calculate the homology class of an oriented transition graph, we can use cuts along the two directions parallel to the unit cell sides. The flow through a cut is defined as the sum of crossings of the oriented transition graph across the cut (taking into account the direction of the crossing). Moreover, the two flow values give the homology class of the oriented transition graph as a vector of two integers. An example is shown in Figure 3.30(b).

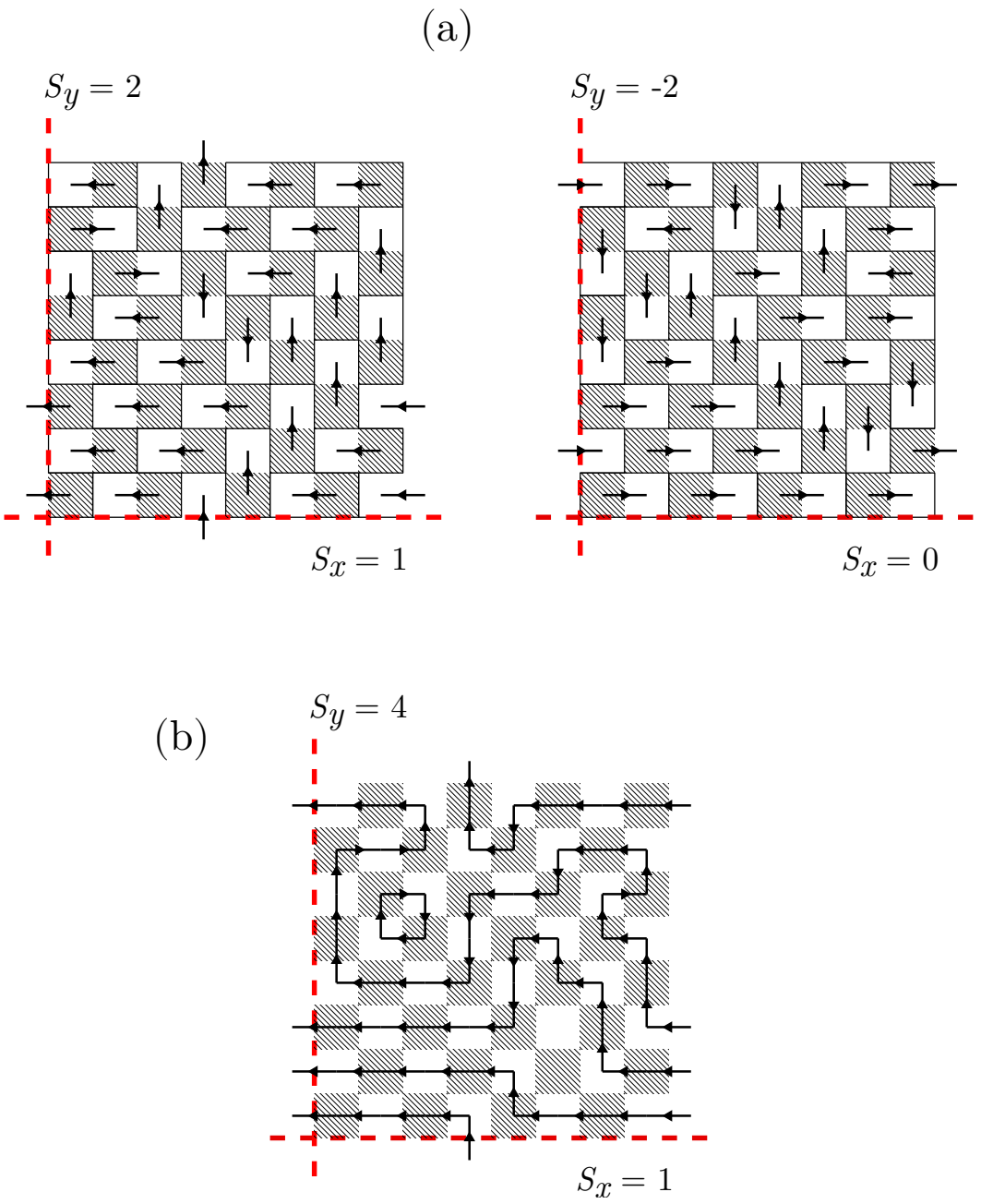


Figure 3.30: Domino configurations on an 8×8 unit cell. (a) Each domino configuration can be assigned to a homology class by calculating the flow across the cuts, giving an integer vector (S_x, S_y). (b) The oriented transition graph for the two domino configurations. These domino configurations are not in the same homology class, since the oriented transition graph is in a nontrivial homology class.

It is important that the cuts are chosen correctly. For a torus, it is fairly straightforward to show that one vertical and one horizontal cut is the correct choice by the following argument. The calculation of flow across cuts gives the correct homology class for a representative set of 1-dimensional chains. For example, a straight horizontal chain gives a flow through the vertical cut, with value equal to the value of the chain, and similar for a straight vertical chain with horizontal cut. Furthermore, any 1-chain that is a boundary contributes zero flow across the cuts. Therefore, the flow through these cuts correctly identifies the homology class for all 1-dimensional chains.

It is also possible to assign a homology class to individual domino configurations. We can use the same method of measuring the flow across cuts, as shown in Figure 3.30(a). Since the oriented transition graph is the difference of two domino configurations, the oriented transition graph is in the trivial homology class when the two individual domino configurations belong to the same homology class. Consequently, two domino configurations are in the same homology class when they are connected by domino flip moves. For the concept of homology class for an individual domino configuration, we use the integer vector (S_x, S_y) to denote flow across horizontal and vertical cuts, which we will refer to as the sector. The sector gives an explicit way to assign domino configurations to homology classes.

By calculating the height function on a unit cell, we get a similar classification. Since the unit cell is not simply-connected, the height function does not necessarily match at either sides of the periodic boundary. In fact, the height change across both sides of the boundary is four times the value of the sector, which we have just introduced. An example domino configuration in a nontrivial sector is shown in Figure 3.31 with the height function overlaid. Furthermore, the height change per plaquette can be given in terms of the sector as

$$\Delta h = \left(\frac{4S_x}{N}, \frac{4S_y}{M} \right), \tag{3.13}$$

where Δh denotes the height change per plaquette. In a planar domino pattern, this corresponds to the mean height change, giving a concept that does not depend on the size of unit cell being considered. Also, the height change per plaquette must have L^1-norm less than or equal to 2 [42]. This means the configurations with $\mid \Delta h_x \mid + \mid \Delta h_y \mid = 2$ are the configurations of extreme height gradient. The domino configurations with extreme height gradient are exactly the set of ladder configurations mentioned in Subsection 2.3.3. This is because each vertical domino in a ladder configuration must give the same height change across it, and similar

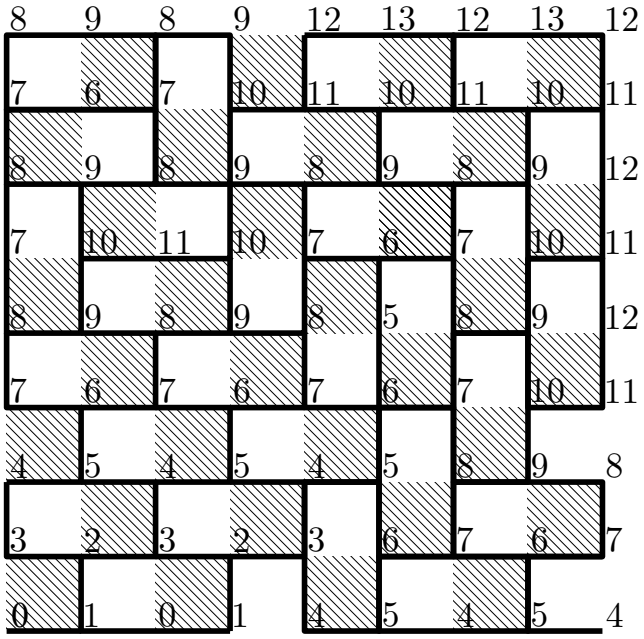


Figure 3.31: Example height function overlaid on a domino configuration of the 8×8 unit cell. The height change per plaquette is $\Delta h = (4, 8)$. This configuration is the same as the one shown on the left of Figure 3.30(a).

for the horizontal dominoes. Therefore, the height change contribution from any domino in the configuration will always be of the same sign, giving maximal mean height change. This allows for an easy identification of ladder configurations.

Chapter 4

Differential Evolution For Parameter Optimisation

The structure and properties of molecular monolayers have been shown to depend on several experimental system parameters [4, 8, 10]. This gives the opportunity to design for particular monolayer structures that have desired properties. Parameters corresponding to the strength and type of interactions between molecules give possibly the best example of this. Therefore, in this chapter, we present our implementation of a differential evolution algorithm to optimise parameters for a given objective property. Explicitly, the interaction parameters define the energy for each configuration, and the temperature along with the energy for each configuration determine the various thermal averaged quantities of the ensemble of interacting polyomino configurations, for example, the internal energy. This will be explained in more detail in section 4.2. Important properties of the system of interacting polyominoes can be described by the set of interaction parameters.

We use the concept of a fitness function to identify configurations that posses desirable properties such as chiral ground state or a sharp heat capacity curve. Generally, this fitness function may not be smooth in any sense, but it will depend only on directly measurable properties of the ensemble of interacting polyomino configurations, which is discussed in section 4.3. The properties of the system of interacting polyominoes depend on our choice of interaction parameters, which can be thought of as a vector in an abstract space. So now we have some (possibly not smooth) fitness function on an abstract vector, and we need to optimise the fitness function by choosing a suitable vector of interaction parameters. For this purpose, we use an optimisation algorithm, described in section 4.4, and discuss results for example choices of the fitness function in sections 4.6, 4.7, and 4.8.

4.1 Freedom Of Energy Assignment

The energy of each polyomino configuration is the dot product of its interaction count vector with the vector of interaction parameters $\vec{n} \cdot \vec{\varepsilon}$. With this expression for the energy of configurations, it is immediately clear that configurations with the same interaction count vector $\vec{n}$ must be energetically degenerate under any choice of the interaction parameters $\vec{\varepsilon}$. For this reason, it will be convenient in several parts of this thesis to use g_i as the degeneracy of configurations corresponding to the ith interaction count vector $\vec{n}^i$. Furthermore, a given choice of interaction parameters $\vec{\varepsilon}$ defines a normal direction for planes in $\vec{n}$-space that are assigned the same energy value. In this sense, we have a necessary degeneracy for configurations corresponding to the same $\vec{n}^i$ and a further degeneracy due to the choice of $\vec{\varepsilon}$.

It is also possible to arrange the set of interaction count vectors $\vec{n}^i$ as ith rows of a matrix $\boldsymbol{A}$, where the order of the rows can be chosen arbitrarily. The list of energies E^i corresponding to each of the $\vec{n}^i$ can be written into a vector of energy values $\vec{E}$. This vector of energy values can be seen to depend on the interaction parameters $\vec{\varepsilon}$ by way of a matrix equation $\vec{E} = \boldsymbol{A} \cdot \vec{\varepsilon}$. Each row $\vec{n}^i$ is generally associated to more than one configuration, with degeneracy g^i. It is simplest to use a matrix $\boldsymbol{A}$ with all rows unique, keeping note of the (energetically degenerate) set of configurations corresponding to each row. Naturally, only interaction count vectors with nonzero degeneracy are included in the matrix $\boldsymbol{A}$. Explicitly writing out a matrix equation clarifies exactly what freedom we have in determining the energies of configurations by altering the interaction parameters; the matrix $\boldsymbol{A}$ maps our free parameters $\vec{\varepsilon}$ to the energies of the interaction count vectors $\vec{E}$. Therefore, the image of the linear map represented by $\boldsymbol{A}$ gives the space of possible solutions for $\vec{E}$. This provides the possible assignments of energy to configurations in full generality, since configurations that correspond to the same $\vec{n}^i$ necessarily have the same energy assignment.

As an example set of polyomino configurations, consider the set of right tromino tilings of a 6×3 unit cell. There are 25 configurations corresponding to 14 unique interaction count vectors. Furthermore, there are 6 interaction parameters. The matrix $\boldsymbol{A}$ for this set of configurations is presented in Equation (4.1). Each of the 14 rows represents a unique interaction count vector. The 6 columns correspond to the different interaction types. Alongside the matrix, the wallpaper groups of corresponding configurations are written, with degeneracy in brackets.

$$A = \begin{bmatrix} 6 & 6 & 0 & 0 & 6 & 6 \\ 4 & 4 & 4 & 4 & 4 & 4 \\ 2 & 2 & 8 & 8 & 2 & 2 \\ 0 & 0 & 12 & 12 & 0 & 0 \\ 12 & 0 & 0 & 0 & 0 & 6 \\ 6 & 0 & 6 & 0 & 0 & 6 \\ 7 & 1 & 3 & 0 & 0 & 6 \\ 9 & 1 & 1 & 0 & 0 & 6 \\ 10 & 2 & 2 & 0 & 0 & 6 \\ 8 & 2 & 4 & 0 & 0 & 6 \\ 6 & 2 & 0 & 0 & 0 & 6 \\ 8 & 2 & 2 & 0 & 0 & 6 \\ 8 & 4 & 2 & 0 & 0 & 6 \\ 6 & 6 & 0 & 0 & 0 & 6 \end{bmatrix} = \begin{matrix} pg(1) \\ p1(2) \\ pg(1) \\ cm(1) \\ p2(2) \\ p2(2) \\ p1(2) \\ p1(2) \\ p2(2) \\ p2(2) \\ p2(2) \\ p1(2) \\ p1(2) \\ p2(2) \end{matrix} \tag{4.1}$$

The form of the matrix $\boldsymbol{A}$ depends on the set of configurations under consideration. Since we will examine various sets of polyomino configurations, some understanding can be gained from the form of $\boldsymbol{A}$ alone. There are several possible interesting cases for the matrix $\boldsymbol{A}$. If there are at least as many interaction parameters as interaction count vectors and all interaction count vectors are linearly independent, then $\boldsymbol{A}$ is surjective, meaning that for any desired vector of energies $\vec{E}$, there exists a choice of interaction parameters $\vec{\varepsilon}$ to achieve it. This is an extreme case that does not occur in practice, since the number of configurations one would like to consider is usually greater than the number of interaction parameters.

The other interesting possibility is when the number of interaction count vectors is at least as great as the number of interaction parameters and there are as many linearly independent interaction count vectors as there are interaction parameters. In this case, $\boldsymbol{A}$ is injective. This would mean there is only (at most) a single choice of interaction parameters $\vec{\varepsilon}$ that gives a specific vector of energies $\vec{E}$. This case is typical, except when additional packing constraints cause the set of configurations to exist within a subspace of the space of interaction counts, as explained in Subsection 3.5.2. For the example in Equation (4.1), the matrix is neither surjective nor injective. In fact, the combination $2n_4 + n_6$ is the same for all configurations, so any energetic contribution that is proportional to this will simply shift the energy values of all configurations.

4.2 Configuration Probabilities

Consider the set of interaction parameters $\vec{\varepsilon}$ as something that an experimentalist has some influence over [8, 10]. For example, suppose it is possible to choose from a selection of molecules with similar shapes but different bonding energies [23, 84] or use a different substrate to alter the relative importance of molecule-molecule interactions [14]. For the systems we are considering, with simple interactions between nearest-neighbours, the matrix $\boldsymbol{A}$ contains the information on what possible assignments can be made to the energy values of each of the configurations by altering the interaction parameters.

Ideally, the experimentalist would have complete freedom, but from the above argument this is generically not the case because the set of energy assignments E^i must be within the image of the matrix $\boldsymbol{A}$. Furthermore, the relative probabilities between configurations are given by Boltzmann weighting, determined by the energies. Therefore, the allocation of probability over the set of configurations can be altered by changing the interaction parameters, but generally there is not complete freedom in how the probabilities are assigned, for the same reason that there is not complete freedom in the set of energy assignments E^i.

Any one of the configurations corresponding to the ith interaction count vector $\vec{n}^i$ will be assigned energy $E^i = \vec{n}^i \cdot \vec{\varepsilon}$. Therefore, the Boltzmann weighting $\exp(-\beta \vec{n}^i \cdot \vec{\varepsilon})$ provides the relative probability for the configuration. The parameter β is the inverse of temperature. However, it is possible to create dimensionless parameters and eliminate β by redefining the interaction parameters $\vec{\varepsilon} \to \vec{\varepsilon}/\beta$ so that $E^i \to E^i/\beta$, and the Boltzmann factor becomes

$$\exp(-\vec{n}^i \cdot \vec{\varepsilon}), \tag{4.2}$$

where the interaction parameters $\vec{\varepsilon}$ and energy of the ith configuration E^i are now defined relative to the temperature. In particular, an increase in the temperature of the system now corresponds to a decrease in the magnitude of the vector $\vec{\varepsilon}$. Therefore, we can still consider the effect of changing temperature by considering a change in the magnitude of $\vec{\varepsilon}$.

4.3 Fitness Functions

Desired system properties can be encoded by using a fitness function of the configurations and their relative probabilities. There are many possible fitness functions that could be chosen, but a natural example is the value of a weighted average of

some quantity that is defined for each configuration. For example, we can consider rotational symmetry. Each configuration has either rotational symmetry or does not. This can be indicated by a Boolean value: 1 meaning the symmetry is present and 0 meaning the converse. Therefore, each configuration has a rotational symmetry value of 1 or 0. The ith interaction count vector $\vec{n}^i$ generally corresponds to several energy-degenerate configurations with different rotational symmetry values. The mean over those configurations gives a value r_i. Furthermore, the weighted average for rotational symmetry over the total set of configurations is

$$\langle R \rangle = \sum_i \frac{1}{Z} r_i g_i \exp(-\vec{n}^i \cdot \vec{\varepsilon}), \tag{4.3}$$

with partition function Z. Note that here we are using a sum over points in $\vec{n}$-space, being careful to include degeneracy g_i and a quantity value r_i that is the average over the configurations corresponding to the same point in $\vec{n}$-space. To find the interaction parameters that promote rotational symmetry, it is necessary to maximise the weighted average in equation (4.3) with respect to the vector of interaction parameters $\vec{\varepsilon}$.

4.4 Differential Evolution Implementation

We have established the space of possible energy assignments, probability assignments, and fitness functions - with the specific example of fitness function $\langle R \rangle$. It is important to identify the requirements for our optimisation algorithm and explain our choice of algorithm on this basis. These are our main requirements:

- The object to optimise is a set of real-valued parameters, i.e. a vector.
- Ideally want minimal computational cost.
- Would like to be able to consider fitness functions for which it is difficult to calculate gradient for.
- Global maxima are most important, rather than local maxima of the fitness function.
- Simplicity is a positive point.
- Rate of convergence should also be prioritised.

Out of the large number of different optimisation algorithms to choose from, we have chosen the genetic algorithm known as differential evolution [92]. The main

reason is that this algorithm is specifically designed to be used on the optimisation of real-valued vector objects. However, the other requirements of our list are also best met by the differential evolution algorithm, compared to other algorihms. Algorithms such as Natural Evolution Strategies [93] and gradient descent methods [94] require the calculation of gradients. This adds computational effort and would restrict our fitness functions to those for which it is feasible to calculate a gradient for. Furthermore, gradient descent methods find local minima, not like the differential evolution algorithm, which finds the global maxima. Many of the usual genetic algorithms represent the object to be optimised as a bit string [95]. This is quite a general way to represent a large set of different problems, but in our case it adds extra complication to the problem compared to the differential evolution algorithm, which provides a specific algorithm for our purpose. There also exist other algorithms similar to the differential evolution algorithm, such as the artificial bee colony algorithm [96]. This algorithm has an advantage over the differential evolution algorithm in terms of exploration of the parameter space, however it loses out on convergence rate and is more complicated to simulate.

To optimise the quantity $\langle R \rangle$ with respect to the interaction parameters, we have chosen differential evolution. This is a form of evolutionary algorithm, which has a population of individuals, a fitness function that can be applied to any of these individuals, and a set of operations for evolving the population forward to the next generation. These operations are mutation, crossover, and selection. Mutation corresponds to the creation of a new trial individual, crossover involves mixing the mutant parameters with that of another individual, and selection occurs to choose fitter individuals for the next generation. These operations increase the fitness of successive generations, leading to a population where all individuals are very similar to the ideal individual whose fitness is a global maximum.

To implement the differential evolution algorithm, we define each individual of the population as a choice for the vector of interaction parameters $\vec{\varepsilon}$. Each choice of $\vec{\varepsilon}$ corresponds to a value of $\langle R \rangle$, which is the fitness of the individual, since it is the quantity that is being optimised. The quantity $\langle R \rangle$ takes a value between 0 and 1, such that individuals with fitness equal to 1 give an optimal choice for $\vec{\varepsilon}$. Our specific example here is the set of configurations made up of straight trominoes with non-chiral interactions generated by a 9×9 periodic unit cell, although we have tested this algorithm on several different polyominoes and cell dimensions. The fully-packed tromino system with non-chiral interactions has 3 relevant interaction types a-b, a-c, and b-c, shown in Figure 4.1.

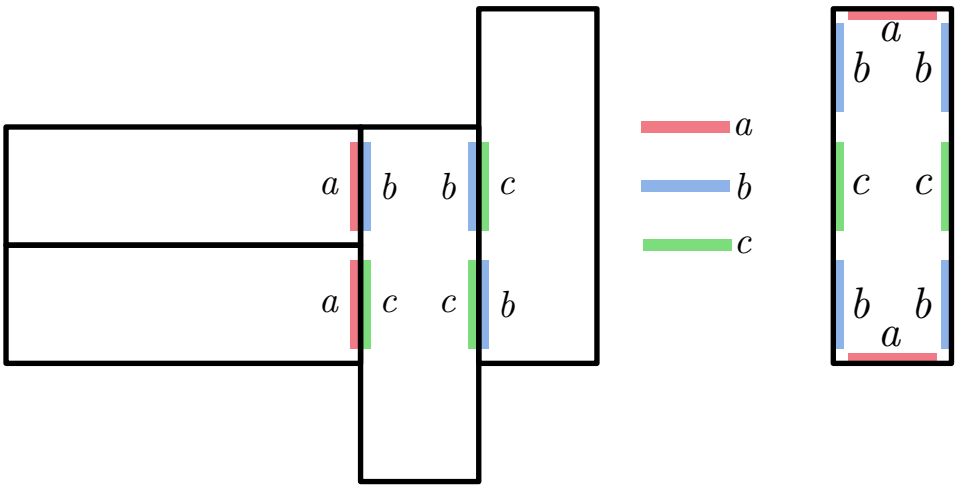


Figure 4.1: Straight tromino with face types a,b,c identified. The 3 interactions a-b, a-c, and b-c are the relevant interactions for the fully-packed system, after taking into account packing constraints.

4.4.1 Algorithm Details

Differential evolution is an evolutionary algorithm, which means successive generations are created with increasing fitness. To create a new generation, each individual is considered in turn. A mutant is spawned from values of the current population and random crossover occurs between the mutant and current individual to make a proposed new individual. The fitness of the new individual is compared with the current individual and one is selected to be used in the next generation. This is iterated over all individuals of the current generation to create the next generation. The three elements of mutation, crossover, and selection can be seen in the pseudocode of Algorithm 1.

For the mutation of individuals, we use the formula given in Ref. [92]. For our implementation, this gives a mutant of the form $\vec{\varepsilon} = \vec{\varepsilon}_1 + F(\vec{\varepsilon}_2 - \vec{\varepsilon}_3)$, where the subscript indices stand for 3 different individuals selected randomly from the current population. For the parameter F, we have tested both $F = 0.8$ and $F = 1$. Both heuristics converge to the same value and within a very similar number of generations, pictured in Figure 4.2. For the crossover, we randomly select the values of each component for the trial individual, which are taken from either the mutant or another (different) random individual, except one randomly chosen component is always taken from the mutant, as suggested in Ref. [92]. The crossover probability to inherit a component's value from the mutant, CR, is another parameter of the algorithm. We tested parameters $CR = 0.2$ and $CR = 0.5$, to give results that are

in agreement, again shown in Figure 4.2. Lastly, the selection of individuals for the new generation is simple. If the fitness of a trial individual is greater than that of the previous individual, replace that individual in the new generation with the trial individual. Otherwise, keep the previous individual for the new generation. As mentioned, every individual is considered before updating to the new generation, meaning it is possible for the entire population to change within one generation, although typically some of the previous generation will survive.

```
1  begin
2      initialise individuals ε⃗_k for k in Population
3      for t in Total Iterations do
4          for i in Population do
5              Select 3 different Individuals x, y, z ≠ i
6              ε⃗_mutant ← ε⃗_x + F(ε⃗_y − ε⃗_z)
7              for j ← 1 to 3 do
8                  if random[0, 1] < CR then
9                      ε^j_temp ⟵ ε^j_mutant
10                 else
11                     ε^j_temp ⟵ ε^j_i
12                 end
13             end
14             Choose a random component j,
15             ε^j_temp ⟵ ε^j_mutant
16             New Generation Individual ⟵ Fitter(ε⃗_temp , ε⃗_i)
17         end
18         Current Generation ⟵ New Generation
19     end
20     Print Generation
21 end
```

Algorithm 1: Differential Evolution

At the first generation we initialise 1000 individuals uniformly within a cubic region from -80 to 80 for each of the 3 interaction parameters to give a significant range of energy values. 1000 individuals is more than is usually necessary for this kind of algorithm, but we have chosen a large number of individuals so that it is possible to effectively visualise a heat map of the fitness function by simply printing the values of the fitness function for all individuals at various times.

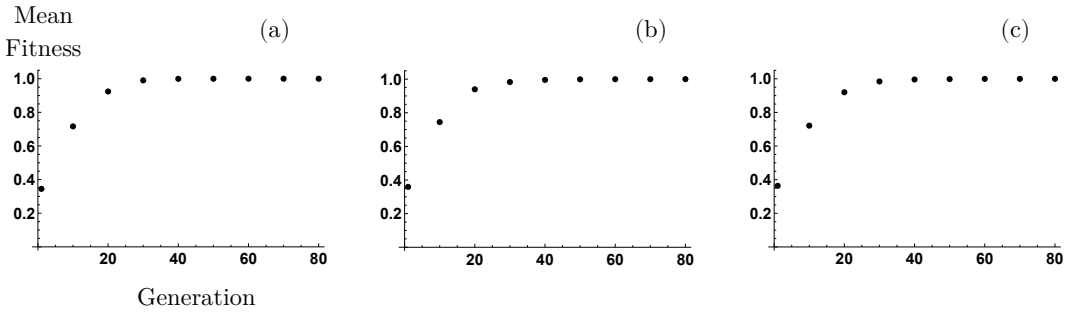


Figure 4.2: Convergence of the mean fitness to the maximum value under the differential evolution algorithm. The interaction parameters are varied within a bounded region of parameter space to optimise for the rotational symmetry of configurations. This was done for several parameters of the differential evolution algorithm, to confirm that the results are robust to the choice of differential evolution parameters F and CR. For (a), $F = 1$, $CR = 0.5$. For (b), $F = 0.8$, $CR = 0.2$. In (c), $F = 1$,$CR = 0.2$.

4.5 First Results For Promoting Configurations With Rotational Symmetry

Using the differential evolution algorithm to find optimal $\vec{\varepsilon}$ parameters for $\langle R \rangle$ gives unhelpful results at first. As can be seen in the left image of Figure 4.3, the individuals are initially distributed uniformly in the cubic region from -80 to 80 in all 3 components. The colour of individuals corresponds to their fitness. There are higher fitness values around the corners $(80, 80, 80)$ and $(-80, -80, -80)$. However, after only 10 iterations of the algorithm, shown in the right hand image of Figure 4.3, many of the individuals spread away from the origin in parameter space. Here, the algorithm is trying to increase the effective energy difference between the configurations with and without rotational symmetry, which causes typical $\vec{\varepsilon}$ vectors to diverge. The large magnitude of $\vec{\varepsilon}$ can also be seen as a quenching of temperature, since the interaction parameters are defined relative to the temperature. The fitness values of the individuals of the algorithm become very close to 1, corresponding to a purple hue seen in Figure 4.3.

The fitness of different individuals becomes difficult to distinguish due to the very great numerical precision that would be required. In this case, the method does not converge on a solution; there are individuals in each octant outside the original cubic region. This still persists even after 100,000 iterations, which is a much greater time than would usually be required. As a fix for this problem, we introduced an allowed region from -80 to 80 in each of the 3 parameters to avoid any of them running to infinity. We don't allow crossover with any mutant that

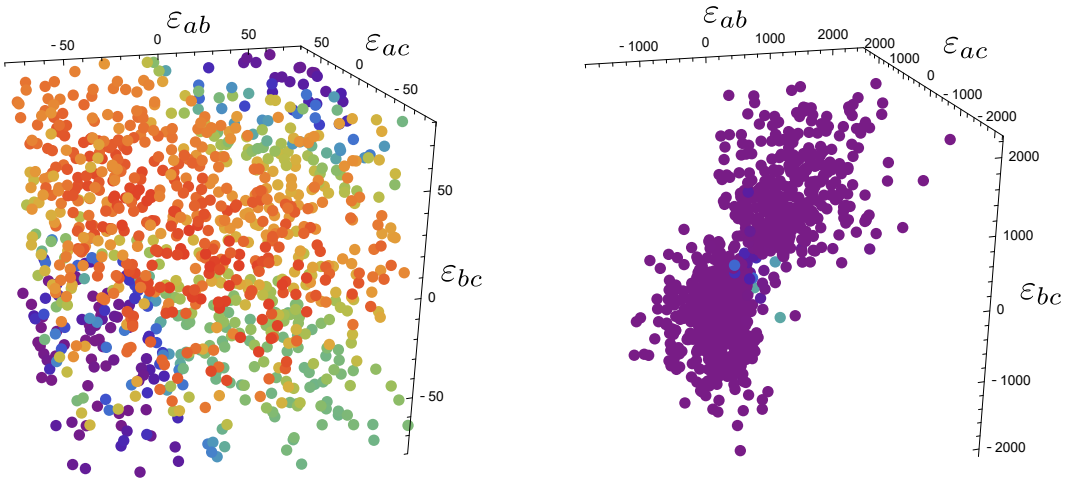


Figure 4.3: Individuals of the differential evolution algorithm represented by points in interaction parameter space. The two images, from left to right, are at generations 1 and 10. The colours correspond to the fitness of the individual, with purple hue being 1, and red being 0.

has fallen outside this region, meaning all our individuals stay within the permitted region. It would also have been possible to fix the magnitude of the vector of interaction parameters $\vec{\varepsilon}$ to a constant, which would allow for optimisation of the relative interaction energies at fixed temperature. However, this approach would miss out on the possibility for the optimisation algorithm to compare solutions of different temperatures, especially the low temperature solutions.

4.6 Promoting Configurations With Rotational Symmetry

The results of the new differential evolution algorithm to maximise the function $\langle R \rangle$ within the permitted region are shown in Figure 4.5, where the individuals of the population are shown as points and their fitness is indicated by their colour. The population of four different generations is displayed for the four figures from left to right, in chronological order. At later times the entire population has converged at the point $(-73.3195, 80, -80)$. This occurs for several runs with the same parameters. This point is on one of the faces of our allowed region, indicating that the fitness function may be more optimal for an individual that is even further from the origin in this direction. This vector of interaction parameters causes the point $(36, 0, 54)$ in $\vec{n}$-space to have the lowest energy value. This point corresponds to two chiral twin configurations, which have rotational symmetry, shown in Figure 4.4. The second

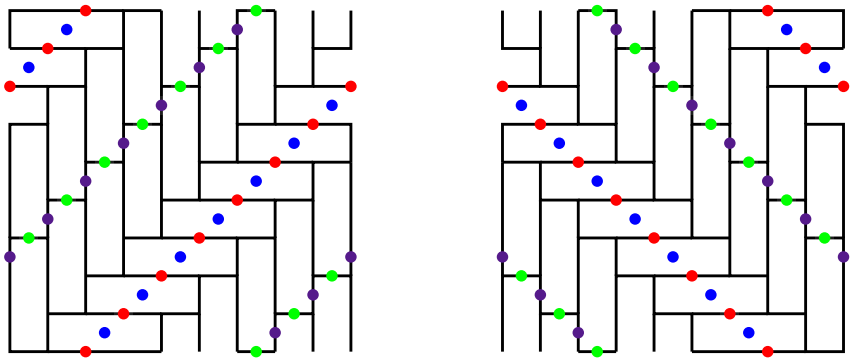


Figure 4.4: The two chiral pair configurations that are lowest energy configurations of the achiral straight tromino system on the 9×9 cell, with interactions$(-73.3195, 80, -80)$. Both configurations correspond to the point $\vec{n} = (36, 0, 54)$, have p2 wallpaper group, and rhombic translational lattice. The 4 non-equivalent rotation centres are shown as differently coloured circles.

lowest energy value is 1960.8 higher than the lowest, a significant amount, meaning that essentially all of the probability is allocated to the two chiral twin configurations with equal lowest energy. Since these two configurations both have rotational symmetry, the fitness value is close to 1 for this choice of interaction parameters. However, by increasing the magnitude of the vector of interaction parameters, the fitness could be further increased to be even closer to 1. The reason this is not possible here is simply because we must restrict our individuals to a finite parameter space.

Taking the magnitude of $\vec{\varepsilon}$ to infinity means we are choosing a particular ground state and reducing the temperature to zero. So what we can learn from the differential evolution algorithm here is that there is a ground state that consists of only rotational configurations, which we should select to optimise the value of $\langle R \rangle$. However, if, for example, we chose the vector of interaction parameters $\vec{\varepsilon} = (80, 80, 80)$, we would select a single rotational configuration to have high probability, so taking the magnitude of $\vec{\varepsilon}$ to infinity gives another fully optimised solution for $\langle R \rangle$. This other solution is no less valid. However, it is not selected by the differential evolution algorithm. In fact, it can be seen in Figure 4.5 that there is a cluster of individuals near this location during the intermediate times, but they eventually all converge at the solution $(-73.3195, 80, -80)$. This behaviour is expected because the eventual solution is the global optimum within the allowed region of parameter space. By putting a bound on the interaction parameters, we can indicate possible solutions of diverging $\vec{\varepsilon}$, but this method does not systematically identify all solutions with diverging $\vec{\varepsilon}$.

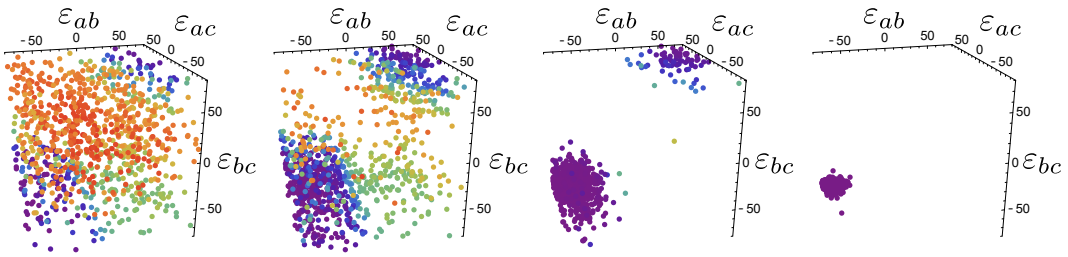


Figure 4.5: Individuals of the differential evolution algorithm plotted as points in interaction parameter space. The images from left to right show generations 1,10,30, and 50. The colours correspond to the fitness of the individual, with purple hue being 1 and red being 0. A metastable cluster appears around the corner $\vec{\varepsilon} = (80, 80, 80)$ until eventually the population converges at $(-73.3195, 80, -80)$.

4.7 Promoting Configurations Without Rotational Symmetry

To further illustrate the point that our problem is best treated by another method, we also consider $1 - \langle R \rangle$ as a fitness function. In other words, we aim to promote configurations without rotational symmetry. The results for this problem are shown in Figure 4.6. In this case, the population converges on the vector of interaction parameters $\vec{\varepsilon} = (-8.0041, -19.0310, 0.7347)$, consistently over different runs of the same program. Since the population has not moved to the boundary of the allowed region, the preferred state is a high-entropy state, where the probability is shared between many configurations. This is not so surprising because strong interactions tend to introduce order into the system, whereas weak interactions give an ensemble of many configurations, some have rotational symmetry and many do not. However, the vector of interaction parameters $\vec{\varepsilon} = (-80, -80, 0)$ would result in the point $\vec{n} = (42, 12, 24)$ taking the lowest energy value, and this point corresponds to two configurations with no rotational symmetry. This means that increasing the vector $\vec{\varepsilon}$ in this direction gives a fitness that tends to 1, which is higher than the fitness achieved by the high-entropy state. So again, the differential evolution algorithm can be used to identify the solutions with finite $\vec{\varepsilon}$, but we are missing the solutions with diverging $\vec{\varepsilon}$.

4.8 Discussion

In both cases here, the differential evolution algorithm gives an answer for the interaction parameters that is most optimal within the finite parameter space, resulting

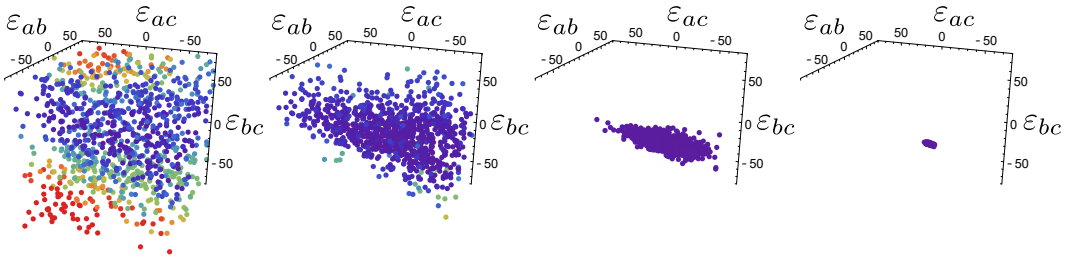


Figure 4.6: The differential evolution algorithm is demonstrated by showing individuals as points in $\vec{\varepsilon}$-space, with the colour corresponding to the fitness of the individual from 0 (red), to 1 (purple). The images shown from left to right are at times 1,10,30, and 70. It can be seen that the ε_{ac} direction takes longer to converge on a single value, because the fitness of individuals varies less strongly in this direction. The population finally converges to $(-8.0041, -19.0310, 0.7347)$.

in an optimised fitness value that is quite close to 1. However, in some cases the $\vec{\varepsilon}$ solution could be more optimal by simply having a larger magnitude, which would be outside our permitted region. In other cases, there are more optimal solutions for different directions of $\vec{\varepsilon}$ that fall outside the permitted region, so we cannot be sure of an optimal solution over the entire parameter space.

If we consider taking away the restriction of the allowed region, the solutions can diverge to large values in parameter space where their fitness values are so close to 1 it becomes infeasible to compare their difference. The differential evolution algorithm is designed to find global maxima at finite parameter values. Since we have found several solutions with infinite parameter values, another approach should be used for our problem. As mentioned before, the solutions with large magnitudes for $\vec{\varepsilon}$ correspond to low temperature solutions. These kinds of solutions occur quite generically over a variety of different polyomino types, meaning that finding a clear picture for these solutions is important for the treatment of the optimisation problem. We can investigate the zero temperature solutions directly, with a separate treatment, rather than attempting to investigate diverging parameter solutions with differential evolution.

Chapter 5

Analysis Of Domino Configurations

In this chapter, we make use of a convex hull construction to identify the low energy states that we found to be important in the previous chapter for giving desirable system properties. We describe how the lowest energy states correspond to boundary components of the convex hull and analyse their properties. We then explain how sets of domino configurations from different unit cells can be compared or combined. Using this principle, we implement a method for enumerating domino configurations with primitive unit cell area up to a given value and we find that a limit shape for the convex hull is reached at primitive unit cell area 8.

Subsequently, we analyse domino configurations using several methods. Kinetic constraints break up the set of domino configurations into separate sectors, quantities of domino configurations are calculated and viewed in $\vec{n}$-space, and degeneracy is used to calculate heat capacity curves. Finally, we calculate the region of parameter space that gives rise to each lowest energy state, giving a sense of how accurately parameters must be tuned to result in a particular lowest energy state.

5.1 Convex Hull Method

We set out here a method for analysing the low temperature states of our interacting polyomino system. Looking at sets of configurations represented as points in $\vec{n}$-space allows for a natural understanding of the possible low temperature states in terms of the interaction parameters. For convenience of notation, we refer to the points in $\vec{n}$-space as $\vec{n}$-vectors.

Our method shares some points in common with that used in Ref. [97] by

Istrail et al. to investigate a model of interacting tiles, where the energy function depends linearly on the number of tiles with particular faces in contact, which they call the tiling statistic. Istrail et al. outline a concept for maximising the energy function by finding extremal values of the tiling statistic. Furthermore, they use this concept to identify energetically favourable motifs of triangles on a hexagonal lattice, where all the triangles have equal shapes but are of different types, having different interactions. We apply a similar concept to polyomino packings by finding extremal values in $\vec{n}$-space. In our model, there are several orientations for each polyomino, and polyominoes of the same shape have identical interactions. Since we have defined our system in this way, each polyomino shape can be thought of as corresponding to a single type of molecule.

As an example for the method of identifying possible ground states, we can consider the set of domino configurations generated on a 4×4 unit cell. The 4×4 unit cell generates 13 domino patterns. Of these, there is 1 with 4-fold rotational symmetry, 11 with 2-fold symmetry, and 1 without rotational symmetry. The interaction count vectors for these configurations happen to lie within a 2-dimensional subspace $n_{ab} = n_{ac}$ of the full 3-dimensional $\vec{n}$ space for dominoes, which is (n_{ab}, n_{ac}, n_{bc}). Therefore, the interaction count vectors can be plotted as points in 2 dimensions. We can use components n_{aa} and n_{bc} because $n_{aa} = 8-(n_{ab}+n_{ac})$, which is due to the packing constraints explained in Subsection 3.5.2. This is shown in Figure 5.1(a). When the interaction count vectors are plotted in this way, we call them the points in $\vec{n}$-space. For a set of configurations generated by a unit cell, the corresponding set of points in $\vec{n}$-space is a finite set, taking integer positions.

Consider the problem of attempting to choose an interaction parameter vector $\vec{\varepsilon}$ so that a particular interaction count vector $\vec{n}^0$ is assigned the lowest energy $E^0 = \vec{n}^0 \cdot \vec{\varepsilon}$, out of all the interaction count vectors in a given set. Note the superscript here is an index for the interaction count vector. Figure 5.1(b) gives an example interaction parameter vector $\vec{\varepsilon}$ for which the interaction count vector on the lower-left is assigned the lowest energy.

Taking the magnitude of the interaction parameter vector to a very large value corresponds to taking the temperature to zero. This results in the lowest energy configurations being assigned all of the probability, as discussed in the previous chapter. Thus, we can address the problem of identifying low temperature solutions from the previous chapter by considering the interaction count vectors that can be assigned lowest energy. Finding an interaction count vector that can be assigned lowest energy and that has desirable properties gives a low temperature solution for an optimisation problem over the interaction parameters.

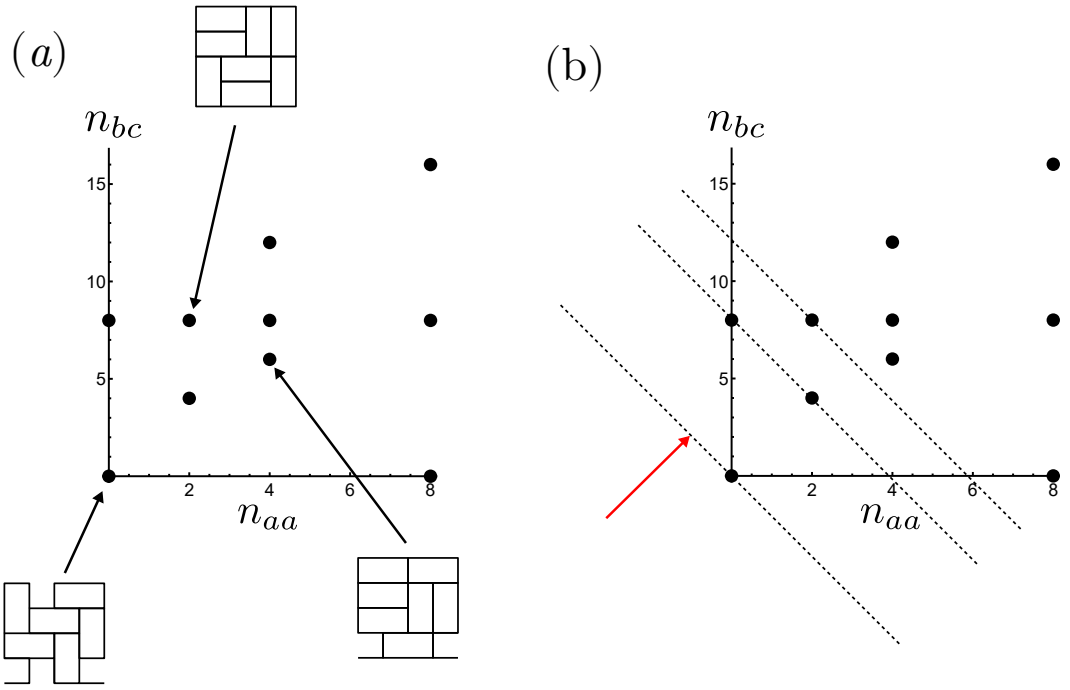


Figure 5.1: Interaction count vectors corresponding to the domino patterns generated by a 4×4 unit cell are plotted as points in $\vec{n}$ space. (a) Some of the domino patterns are indicated. Note that in this case each of the 3 indicated domino patterns are unique to their $\vec{n}$-vector. This is not the case for general configurations. (b) An example interaction parameter vector $\vec{\varepsilon}$ is shown by the red arrow, and dashed lines correspond to hyperplanes of equal energy under the example $\vec{\varepsilon}$ vector.

To determine the conditions for $\vec{n}^0$ to have the lowest energy over the points of $\vec{n}$-space, we can think in terms of hyperplanes perpendicular to $\vec{\varepsilon}$. An example is shown in Figure 5.1(b). These hyperplanes are equal-energy subspaces for the $\vec{n}$-space. Putting a hyperplane through an interaction count vector $\vec{n}^0$ with normal in the direction of the $\vec{\varepsilon}$ vector, the points in $\vec{n}$-space on one side of the hyperplane have higher energy than $\vec{n}^0$, and the points in $\vec{n}$-space on the other side of the hyperplane have lower energy than $\vec{n}^0$. Therefore, to have an interaction count vector $\vec{n}^0$ be assigned the lowest energy, there must exist a hyperplane through $\vec{n}^0$ for which all other points of $\vec{n}$-space are strictly on one side of the hyperplane. This hyperplane determines the normal of an $\vec{\varepsilon}$ vector that assigns the lowest energy to $\vec{n}^0$. In the example of Figure 5.1(b), the vector $\vec{n}^0$ is on the lower-left, and the hyperplane passing through it has all other points of $\vec{n}$-space on one side of it.

For a hyperplane to separate one interaction count vector $\vec{n}^0$ from the set of interaction count vectors, the point $\vec{n}^0$ must be a vertex of the convex hull of that set. The convex hull of a set of points in $\vec{n}$-space is the minimal convex region that includes that set of points. Any hyperplane that separates one point from the other points must only touch the convex hull at that one point. Therefore, the point $\vec{n}^0$ must be a vertex of the convex hull of points in $\vec{n}$-space. Using Figure 5.1(a) as an example, the configuration at point $(n_{aa}, n_{bc}) = (4, 6)$ is within the convex hull of points and cannot be assigned lowest energy. Visually, any interaction parameter vector $\vec{\varepsilon}$ will fail to make this point the lowest energy because it is surrounded by other points, at least one of which will take lower energy value.

It is the vertices of the convex hull of interaction count vectors that can be made to have the lowest energy. Furthermore, the domino configurations that can be made unique energetic lowest energy states are those corresponding uniquely to a vector $\vec{n}$ that is a vertex of the convex hull. The convex hull for the interaction count vectors of the 4×4 unit cell is a quadrilateral, so there are four vertices, depicted as red points in Figure 5.2. The corresponding configurations happen to be unique to their position in $\vec{n}$-space and are pictured in Figure 5.2, having wallpaper groups pmm, p4g, pgg, and cmm (going anticlockwise from upper-right in Figure 5.2). These 4 configurations are therefore the possible non-degenerate lowest energy states. For example, the interaction parameter vector $\vec{\varepsilon} = (\varepsilon_{ab}, \varepsilon_{ac}, \varepsilon_{bc}) = (-1, -1, -1)$ gives the p4g configuration as unique lowest energy state. In this case, the lowest energy state has 4-fold rotational symmetry and primitive unit cell of area 8. However, if instead the interaction parameter vector is $\vec{\varepsilon} = (-1, -1, -2)$, the lowest energy state will be made up of 3 configurations, including the p4g configuration, which is now degenerate under this choice of interaction parameters.

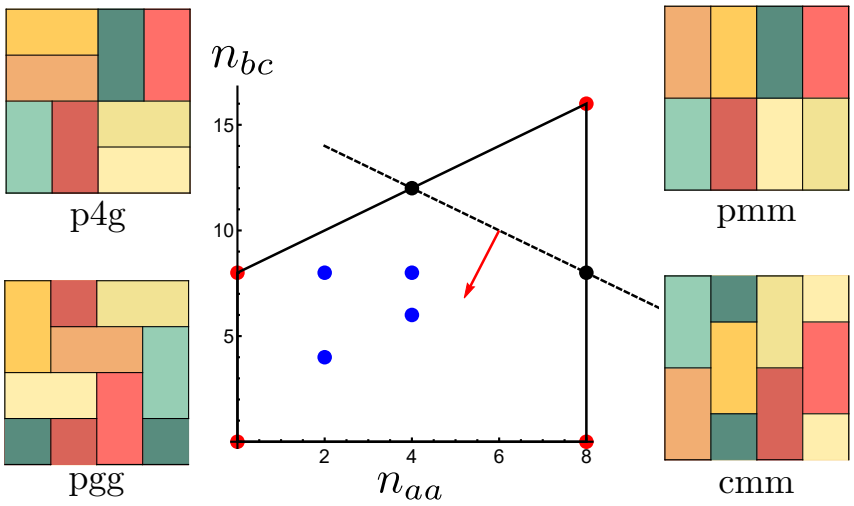


Figure 5.2: Configurations of dominoes on the 4×4 unit cell. Each point corresponds to the interaction count vector of at least one configuration, and the boundary of the convex hull is shown as dark lines. Vertex points are red, other boundary points are black, and interior points are blue. The red arrow gives an example vector of interaction parameters, such that the top-right configuration is the ground state. The dashed line indicates a hyperplane of equal energy for the example interaction parameter vector, and passes through the configurations of second lowest energy. Also shown are illustrations for the vertex configurations, along with their wallpaper group. The colours used for drawing dominoes are purely illustrative.

The red arrow in Figure 5.2 is an example vector for $\vec{\varepsilon}$. In this case, the least value of $E = \vec{n} \cdot \vec{\varepsilon}$ corresponds to the configuration in the upper-right. Furthermore, the two black points on the boundary of the convex hull shown in Figure 5.2 (close to the upper-right) constitute the domino configurations of second lowest energy. These points are iso-energetic because they have equal scalar projection onto the example $\vec{\varepsilon}$ vector. To calculate vertex points and other points on the boundary of the convex hull, we use the Qhull library that was created by C. Bradford Barber and Hannu Huhdanpaa at the Geometry Center and Harvard University. This is important for the larger sets of points that are difficult to calculate by hand. Detail of the algorithm can be found in Ref. [98].

Since the domino patterns on the 4×4 unit cell have interaction count vectors that fall within the plane $n_{ab} = n_{ac}$, each pattern must be assigned an energy equal to its mirror image (chiral twin) pattern. If a pattern has interaction count vector (n_{ab}, n_{ac}, n_{bc}), its chiral twin has interaction count vector (n_{ac}, n_{ab}, n_{bc}) because each *a*-*b* interaction becomes an *a*-*c* interaction after the mirror is applied to form the chiral twin. Due to the fact that dominoes on the 4×4 unit cell have $n_{ab} = n_{ac}$, it is not possible in this case to favour one configuration over its mirror image energetically. Because of this, we will call $n_{ab} = n_{ac}$ the achiral plane. Although it does contain chiral patterns generally, they cannot be distinguished energetically.

5.2 Chiral Interactions

In the general case, the set of domino configurations define a 3D region in the $\vec{n}$-space. For example, the convex hull for $N = M = 8$, which is shown in Figure 5.3, has 1,224,518 domino patterns, divided into 1,551 different points in $\vec{n}$-space. Patterns that are generated on the 4×4 unit cell can also be generated on the 8×8 unit cell because the unit cells are commensurate. The patterns generated by the 4×4 unit cell form a subset of the patterns on the achiral plane. The 8×8 unit cell also generates configurations with $\vec{n}$-vectors that are not on the achiral plane. These configurations must be chiral patterns, in the sense introduced in Subsection 2.2.4, i.e. they must have wallpaper group which is either p1, p2, or p4. This is because applying a mirror to the configuration turns *a*-*b* interactions into *a*-*c* interactions. However, applying a rotation or translation to the configuration does not change the $\vec{n}$-vector. Therefore, if $n_{ab} \neq n_{ac}$, the reflected configuration must be inequivalent to a rotation or translation, so the pattern is chiral.

In comparison to the 4×4 unit cell, the majority of configurations on the 8×8 unit cell do not correspond to vertices of the convex hull. Out of the 1,224,518

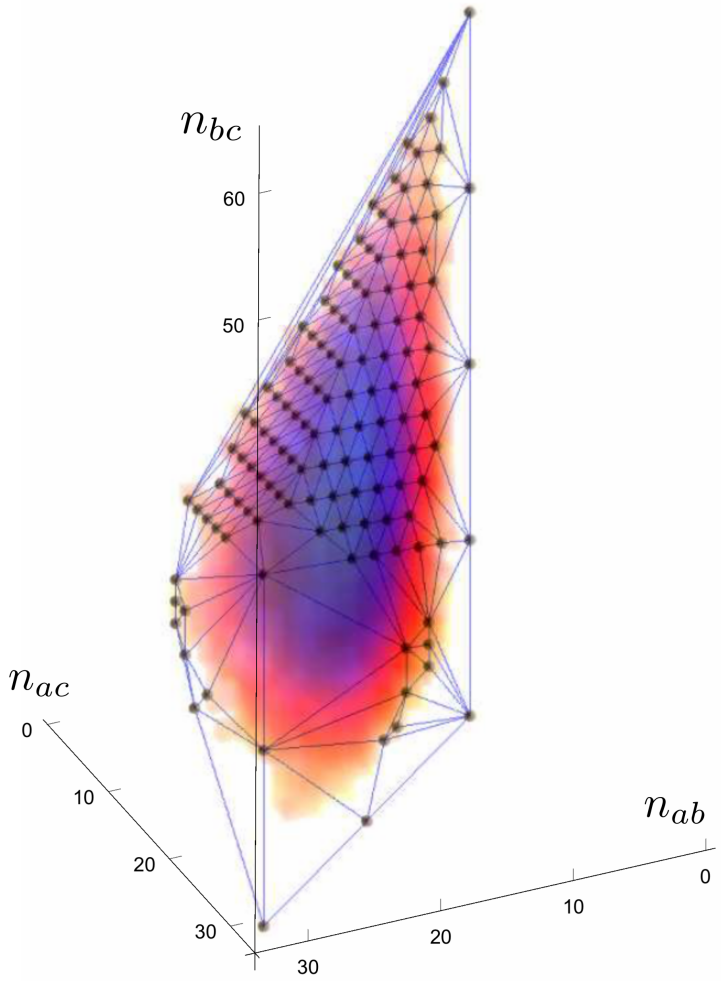


Figure 5.3: Points correspond to the configurations in $\vec{n}$-space which exist on the boundary of the convex hull of the set of configurations which can be made by dominoes on an 8×8 unit cell. The vertex points on the achiral plane are the same as for the $N = M = 4$ case. Away from the achiral plane, the convex hull has 12 vertices corresponding to 24 configurations - 6 p1 and 18 p2. Configurations not on the achiral plane have cyclic (chiral) point groups. Also, the log of density of states is visualised as a heat map, with dark colours signifying high density.

domino patterns, there are only 16 vertices, corresponding to 28 domino patterns. This is a strong limitation to the possible choices for the design of unique ground states. However, since we have an enumeration of states, we can design for desired properties by choosing the ground state that has the most favourable properties. For example, on the 8×8 unit cell, the vertex state $\vec{n} = (32, 32, 32)$ corresponds to exactly one configuration, which has 4-fold rotational symmetry. It is possible to design for 4-fold symmetry by choosing an appropriate combination of interaction energies to make this configuration the unique energetic ground state. One such choice would be $\vec{\varepsilon} = (-1, -1, -1)$. Conversely, one could design for a ground state with no rotational symmetry by finding a vertex state which corresponds to only configurations without rotational symmetry. For the 8×8 unit cell, there are only two such vertex states: $\vec{n} = (34, 20, 24)$ and its chiral twin $\vec{n} = (20, 34, 24)$. The permitted periodicity of the configuration is important to this problem. For the 4×4 unit cell, all ground states have some rotational symmetry; therefore, it is not possible to design for a ground state without rotational symmetry in that case.

The degree of handedness of a domino packing can be given a qualitative value using the distance from the achiral plane, defined as $|n_{ab} - n_{ac}|/\sqrt{2}$. Chiral twin configurations correspond to positions of $\vec{n}$-space that have the achiral plane as midpoint between them. Thus, if a configuration corresponds to a point that is far from the achiral plane, its chiral twin will be quite dissimilar to it, in regard to the count of interaction types. To design an extremally chiral configuration, the vector of interaction energies should have the direction orthogonal to this plane, $\vec{\varepsilon} = (-1, 1, 0)$, so that there is a strong energetic preference for one type of chiral interaction compared to the other.

5.3 Achiral Subspace

In the preceding subsection, we described interactions that strongly favour one chirality over the other. In the converse case, if the interaction parameter vector lies within the achiral plane, each configuration is assigned equal energy to its chiral twin. Therefore, this is the relevant subspace of interaction parameters for a system of molecules that is known to have interactions that do not break the reflection symmetry of the molecular tiling.

When considering the interaction parameter vector being restricted to the subspace of the achiral plane ($\varepsilon_{ab} = \varepsilon_{ac}$), we can project points in $\vec{n}$ space onto the achiral plane to obtain an effective $\vec{n}$-space for the reduced problem. This is because points that are separated by a displacement perpendicular to the achiral plane must

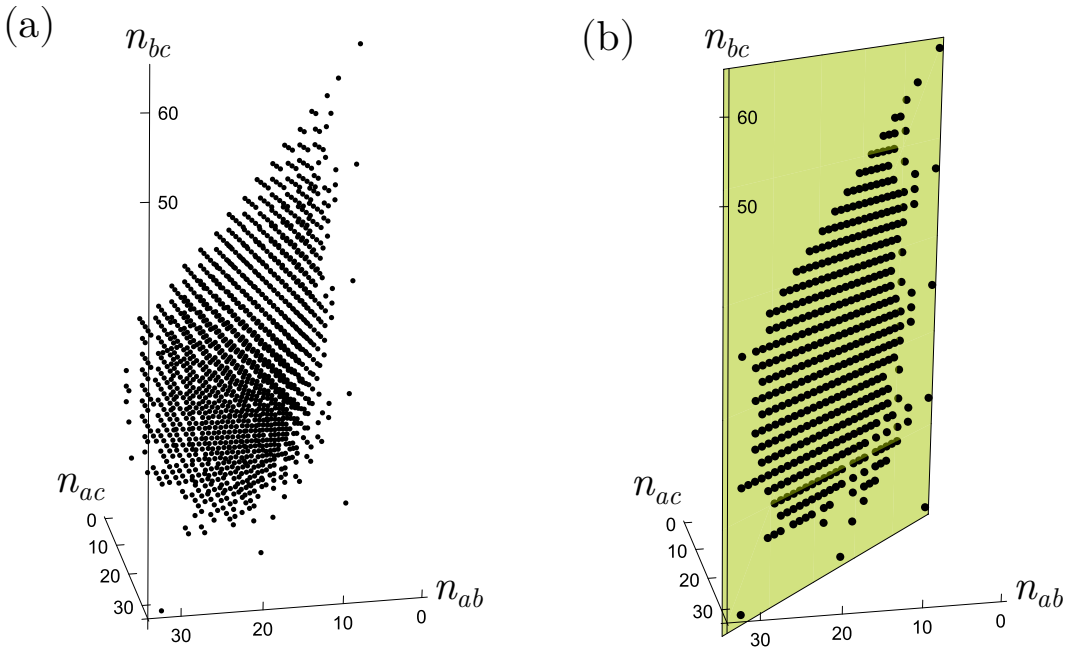


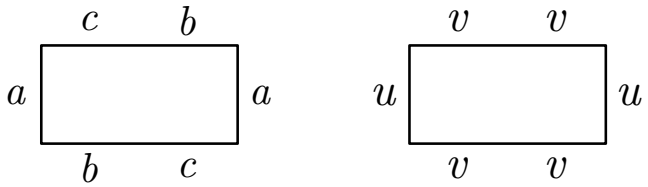
Figure 5.4: (a) The points in $\vec{n}$-space corresponding to domino patterns generated on the 8×8 unit cell. (b) The points projected onto the $n_{ab} = n_{ac}$ achiral plane. This gives the effective $\vec{n}$-space when the interaction parameter vector $\vec{\varepsilon}$ is constrained to $\varepsilon_{ab} = \varepsilon_{ac}$, which means the interaction does not favour one interaction over its chiral twin.

be assigned equal energy by any $\vec{\varepsilon}$ vector that lies in the achiral plane. This concept is shown for the domino configurations of the 8×8 unit cell in Figure 5.4.

We can consider a further subspace by defining the domino face types to respect both the rotational and reflection symmetry of the domino shape. This is the convention that we adopted for the other polyomino shapes, discussed in Subsection 3.5.1. For this new definition of face types, we will use the term achiral domino, since its face types respect the reflection symmetry of the domino shape. We use face types denoted u and v, shown in Figure 5.5, and obtain the packing constraints

$$\begin{aligned} \frac{2NM}{2} &= 2n_{uu} + n_{uv}, \\ \frac{4NM}{2} &= 2n_{vv} + n_{uv}. \end{aligned} \tag{5.1}$$

Since there are $Q = 2$ face types, we are left with $Q(Q-1)/2 = 1$ interactions in the reduced interaction count vector. However, instead of choosing (n_{uv}) as our interaction count vector, we choose (n_{uu}), since this is simply equal to n_{aa} from the more



$$\begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 \end{pmatrix} \begin{pmatrix} n_{aa} \\ n_{ab} \\ n_{ac} \\ n_{bb} \\ n_{bc} \\ n_{cc} \end{pmatrix} = \begin{pmatrix} n_{uu} \\ n_{uv} \\ n_{vv} \end{pmatrix}$$

Figure 5.5: Upper-left is the chiral domino, and upper-right is the achiral domino. Shown below is the matrix conversion for the full interaction count vectors of the two definitions of face types.

general domino interactions. Because the interaction count vector for the achiral domino has only 1 component, the vertices of the convex hull simply correspond to the two extremal values of n_{aa}. The maximum value for n_{aa} corresponds to configurations where all dominoes are aligned in the same direction, possibly with rows shifted with respect to each other. The minimum value for n_{aa} corresponds to a more complicated set of configurations. There are no other lowest energy states for the achiral domino, which motivates our decision to explore the parameter space for the more general domino interactions that allow face types to disobey the reflection symmetry of the domino shape.

5.4 Convex Hull Boundary

For a generically oriented interaction parameter vector $\vec{\varepsilon}$, the lowest energy configurations are those corresponding to a single vertex of the convex hull in $\vec{n}$-space. However, for $\vec{\varepsilon}$-vectors with finely-tuned direction, it is possible to make other components of the boundary of the convex hull be the lowest energy state. Since the interaction count vector for a domino configuration is 3-dimensional, these other components can either be faces or edges of the convex hull. When the interaction parameter vector $\vec{\varepsilon}$ is perpendicular to a face of the convex hull and is pointing inwards towards the convex hull, that face is the lowest energy state, meaning all configurations corresponding to $\vec{n}$-vectors on that face will take the lowest energy

value. For edges of the convex hull, an analogous statement is true. Edges allow a 1-dimensional freedom in the $\vec{\varepsilon}$-vector that causes them to be lowest energy state, whereas for faces, the $\vec{\varepsilon}$ vector is completely determined by the normal of the face.

In terms of design of lowest energy states, if an $\vec{n}$-vector lies on an edge (or face) of the convex hull and is not a vertex of the convex hull, it can be assigned the lowest energy, but only if the other configurations on that edge (or face) are also assigned lowest energy, which leads to a degenerate ground state. Each vertex, edge, and face of the convex hull are boundary components of varying dimensions. Therefore, it is the boundary components that define the possible lowest energy states. We also use the term boundary components when analysing convex hulls in higher than 3 dimensions.

For the domino patterns generated on the 8×8 unit cell, the boundary of the convex hull is shown in Figure 5.6. There are 12,237 boundary configurations, meaning that about 1% of configurations of the 8×8 unit cell can be designed as (possibly degenerate) ground state. Some of the boundary components correspond to special subsets of domino configurations. The line $n_{ab} = n_{ac}, n_{bc} = 0$ consists of the configurations with maximal L^1 norm of the sector, thereby corresponding to the set of ladder configurations described in Subsection 2.3.3.

Another notable component of the boundary of the convex hull is the edge $n_{ab} = n_{ac} = 0$, which has $n_{aa} = 64$ because of the packing constraints. Since these configurations have maximal a-a interactions, this edge corresponds to the set of configurations that have dominoes of only one orientation. A typical configuration is shown in Figure 5.7 This set of configurations can be assigned the lowest energy by a range of possible interaction parameters of the form $\vec{\varepsilon} = (\cos(\theta+\pi/4), \sin(\theta+\pi/4), 0)$ with θ from roughly $-72°$ to $72°$. The number of configurations corresponding to this ground state is of order the length of the system, since each row of dominoes can be shifted independently of the others.

To design for a very high degeneracy ground state, it is necessary to have precise control of the interaction parameters. A striking feature of the boundary of the convex hull for $N = M = 8$ are the faces with many distinct states, which correspond to ground states with especially high degeneracy for energy vectors $\vec{\varepsilon}$ perpendicular to either of these faces. There are two such faces, distinguished by their chirality. They can be seen as the faces containing many points in Figure 5.6. The equation for the normal of one of these faces is $NM = n_{ac} + n_{bc}$, meaning that the direction of the vector of interaction parameters has zero freedom. This is in contrast to the ground state corresponding to the edge $n_{ab} = n_{ac} = 0$, which can be established by a range of possible interaction parameters.

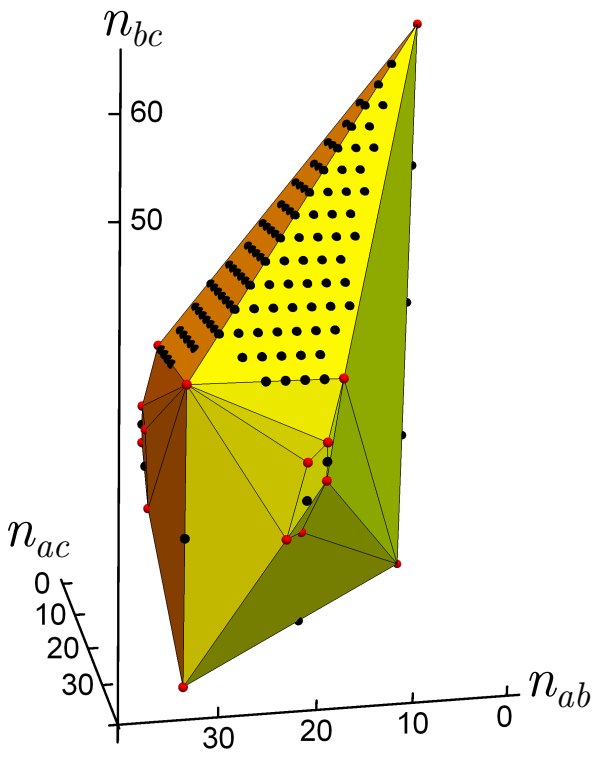


Figure 5.6: The boundary of the convex hull of interaction count vectors (points in $\vec{n}$-space), for domino patterns generated on the 8×8 unit cell. Red points correspond to vertices of the convex hull, while black points correspond to other points in $\vec{n}$-space that are part of the boundary. There are 130 boundary points, and of these, 16 are vertices.

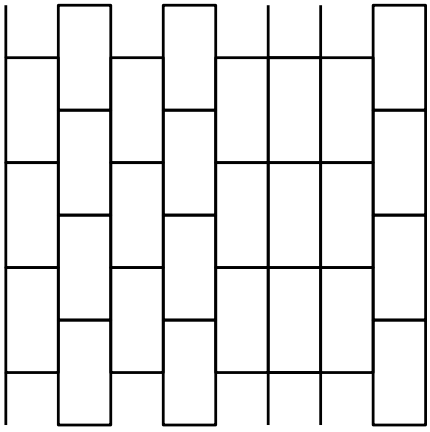


Figure 5.7: Typical domino configuration on the 8×8 unit cell with all dominoes oriented in one direction. Note that $n_{ab} = n_{ac} = 0$.

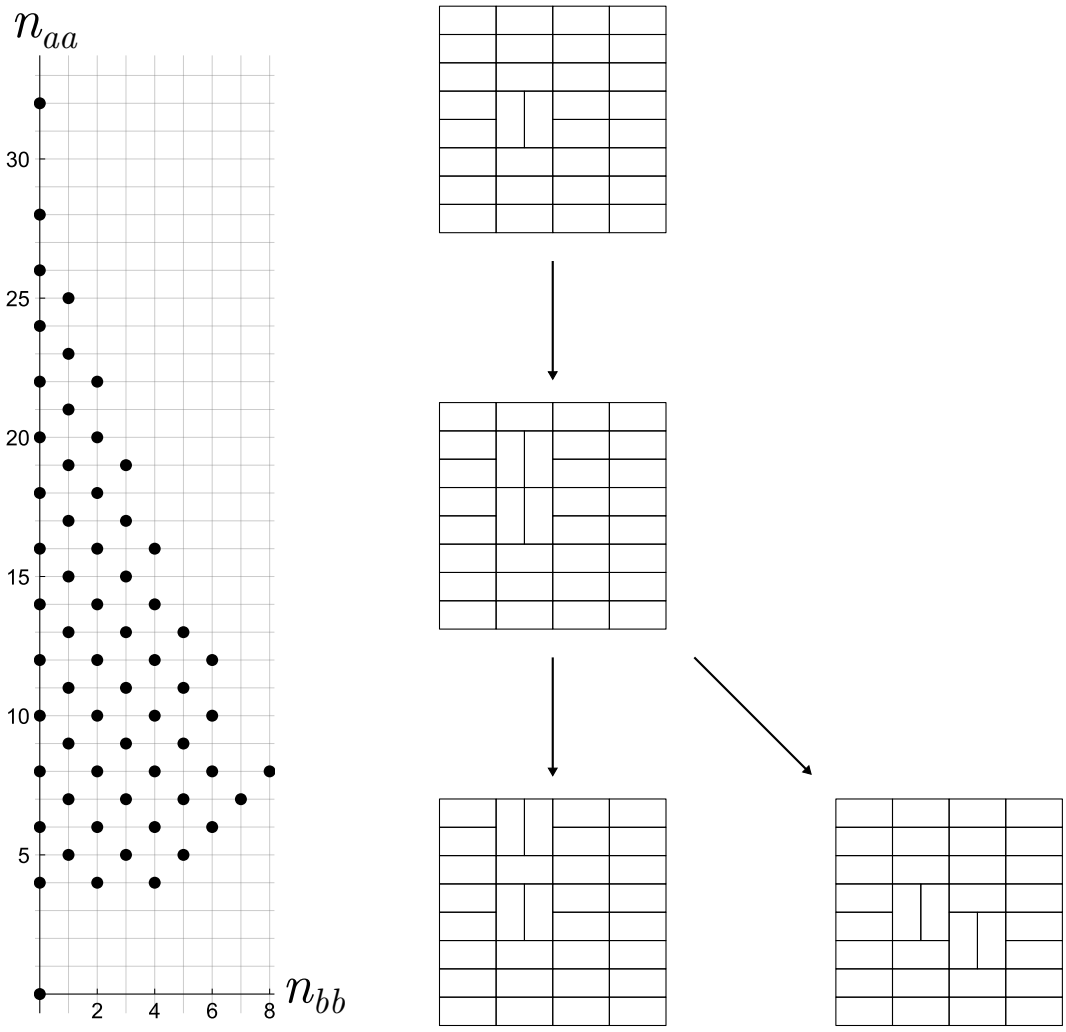


Figure 5.8: One of the two highly degenerate faces of the convex hull for dominoes on the 8×8 unit cell is shown. On the left, the points in $\vec{n}$-space that reside on the face are plotted. Note that coordinates (n_{bb}, n_{aa}) are more convenient than (n_{ab}, n_{ac}, n_{bc}) because $NM - n_{ac} - n_{bc} = 2n_{cc} = 0$ for this face. On the right, it is possible to make domino flips that create *b*-*b* interactions, without creating *c*-*c* interactions. In this way, the face of the convex hull can be traversed by domino flips.

The combined number of domino patterns corresponding to the two high degeneracy faces is 9,916, which makes up roughly 81% of the boundary domino patterns. Each point in $\vec{n}$-space on either of these two faces corresponds to at least one domino configuration from the trivial sector $(0, 0)$. The sectors, as defined in Section 3.6, correspond to the sets of domino configurations that are connected by domino flip moves. Therefore, the trivial sector of the ground state remains highly degenerate, even when considering kinetic constraints, and the high degeneracy faces can be traversed by domino flips, as illustrated in Figure 5.8.

5.5 Unit Cell Comparison

So far, we have only considered sets of domino configurations generated by single unit cells. However, the sets of domino configurations generated by different unit cells can be compared as planar patterns. To understand periodic planar domino patterns, it is important to consider the patterns generated by various unit cells. Different unit cells generate some of the same planar periodic domino patterns, but not all. For example, the convex hulls in $\vec{n}$-space for the 6×6 and 8×8 unit cells are shown in Figure 5.9. The convex hull has a different shape, depending on the patterns generated, and the patterns at vertices of the convex hull are generally not the same. On the 6×6 unit cell, the vertex point $\vec{n} = (12, 24, 12)$ corresponds to a planar domino pattern (shown in Figure 5.10(a)) that cannot be generated by the 8×8 unit cell. Similarly, the pattern in Figure 5.10(b) can be generated on the 8×8 unit cell with the interaction count vector $\vec{n} = (32, 32, 32)$, but cannot be generated by the 6×6 unit cell.

It is possible to combine the sets of domino patterns from several unit cells to enumerate low periodicity patterns more effectively. As previously explained in Subsection 3.4.2, it is possible to generate all patterns with primitive unit cell area less than or equal to a given value A by using the patterns generated by the square cells with side lengths from $A/2$ up to A. For example, the 6×6 and 8×8 unit cells generate all domino patterns with primitive unit cells of area up to 8. To be able to generate all these patterns with a single unit cell, we would have to use the 24×24 unit cell, which is computationally intractable using algorithm DLX.

To compare the interaction count vectors from configurations generated by different unit cells, we must use values per area, rather than per unit cell. Dividing the $\vec{n}$-vector by the area of the unit cell would introduce floating-point numbers and require testing of equality between floating-point numbers. To avoid this, we transform the interaction count vectors to integer valued vectors that are proportional to

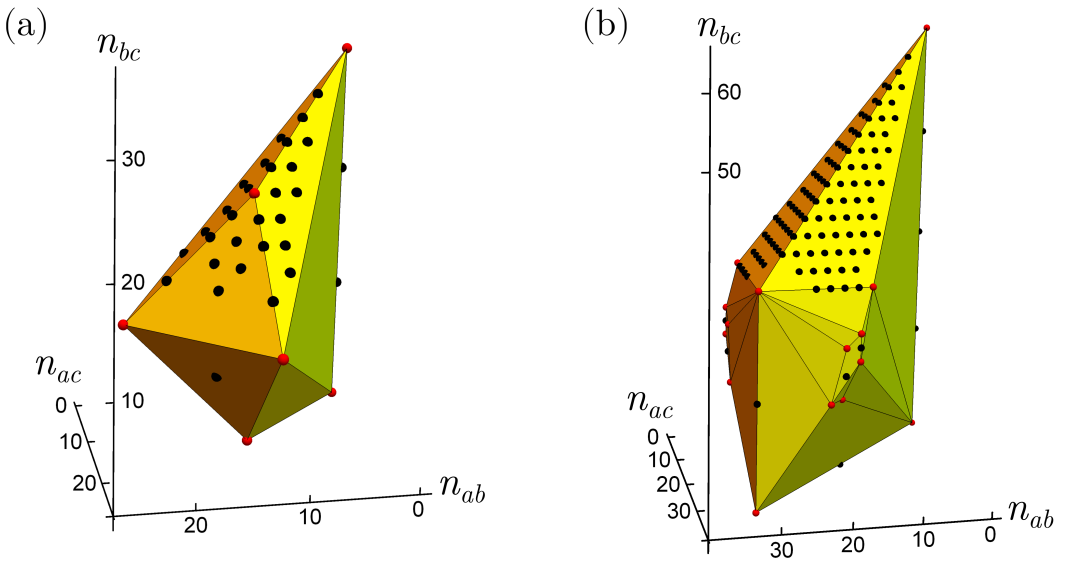


Figure 5.9: Convex hulls in $\vec{n}$-space for the set of domino configurations generated by (a) the 6×6 and (b) the 8×8 unit cells. For (a), there are 6 vertices corresponding to 16 vertex configurations having p1, p2, pmm, pmg, and cmm wallpaper groups, and there are 92 boundary configurations. Whereas for (b), there are 16 vertices with 28 vertex configurations having p1, p2, pmm, pmg, cmm, and p4g wallpaper groups, and there are 12,237 boundary configurations.

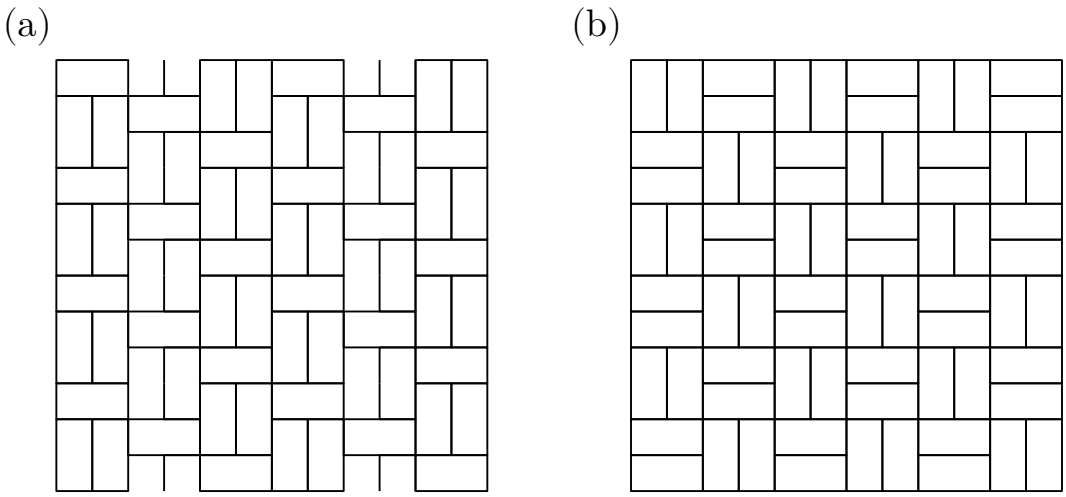


Figure 5.10: (a) Domino pattern with p2 wallpaper group and primitive unit cell area 6. This pattern can be generated by the 6×6 unit cell with interaction count vector $\vec{n} = (12, 24, 12)$. (b) Domino pattern with p4g wallpaper group and primitive unit cell area 8. This pattern can be generated by the 8×8 unit cell with interaction count vector $\vec{n} = (32, 32, 32)$.

the values of the interaction count vectors per area. For example, to compare the configurations generated by the 6×6 and 8×8 unit cells, we multiply the interaction count vectors of the 6×6 unit cell by 16, and we multiply the interaction count vectors of the 8×8 unit cell by 9. Conceptually, one can imagine the configurations being placed in the larger 24×24 unit cell, then comparing the interaction count vectors. This is possible because the 24×24 unit cell is commensurate with both the 6×6 and 8×8 unit cells. Generally, to compare configurations of an $x \times x$ and $y \times y$ unit cell, the interaction count vectors from the $x \times x$ unit cell must be multiplied by $y^2/\gcd(x,y)^2$, and those from the $y \times y$ unit cell must be multiplied by $x^2/\gcd(x,y)^2$. This is the smallest multiplication necessary to give a proper comparison between integer valued vectors.

5.5.1 Enumeration By Primitive Unit Cell Area

By combining domino patterns from multiple unit cells, we obtain the convex hulls in $\vec{n}$-space for domino patterns with primitive unit cell area up to 2, 4, 6, and 8, shown in Figure 5.11. Listing the convex hulls in this way is more natural than using the convex hulls of all patterns generated by progressively larger unit cells, since the primitive unit cell area is a basic property of the planar domino patterns. In other words, we enumerate planar domino patterns in order of increasing spatial periodicity.

Unit cells with side lengths from $A/2$ to A generate all domino patterns with primitive unit cell area up to A. But, square unit cells of odd side length are not needed because dominoes cannot be fully packed in them. Furthermore, we filter the domino configurations by their primitive unit cell area to obtain only the patterns with primitive unit cell area up to the specified value. For example, to generate all the domino patterns with primitive unit cell area up to 6, we generate configurations on 4×4 and 6×6 unit cells and only keep the configurations with primitive unit cell area 2, 4, or 6.

In Figure 5.11, we combine sets of domino configurations from several square unit cells. To be able to meaningfully compare the interaction count vectors from configurations generated by these unit cells, we multiply the original interaction count vectors by values 144, 36, 16, and 9 for unit cells 2×2, 4×4, 6×6, and 8×8 respectively. For this reason, the interaction count vectors shown in Figure 5.11 are equal to the values that the domino configurations would have on a 24×24 unit cell.

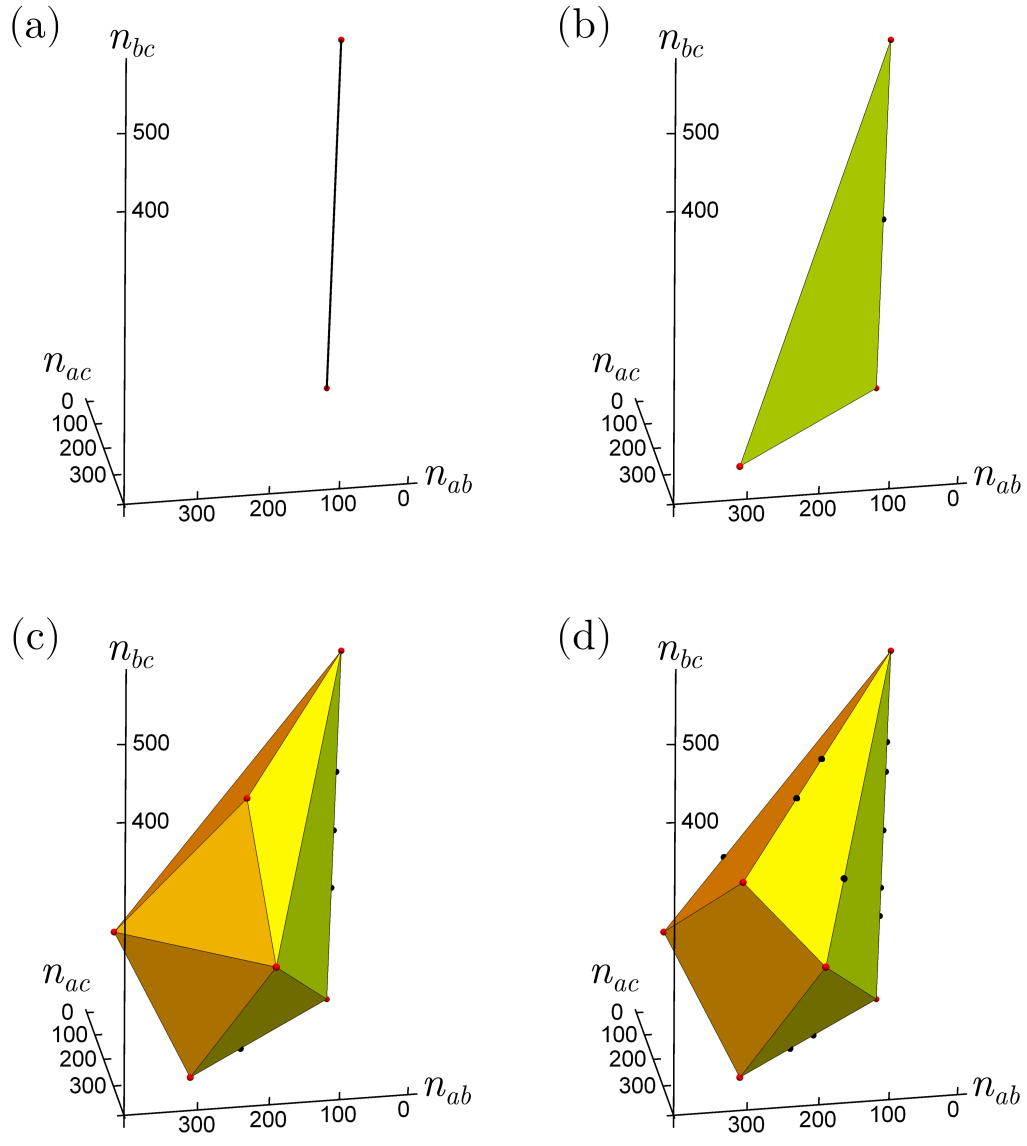


Figure 5.11: The convex hulls of points in $\vec{n}$-space corresponding to domino patterns with primitive unit cell area up to 2, 4, 6, and 8 are shown in (a), (b), (c), and (d) respectively. The number of vertices in each case is 2, 3, 6, and 6. The total number of configurations for each is only 2, 4, 12, and 27, meaning there are 2, 2, 8, and 15 domino patterns with primitive unit cell area 2, 4, 6, and 8, respectively.

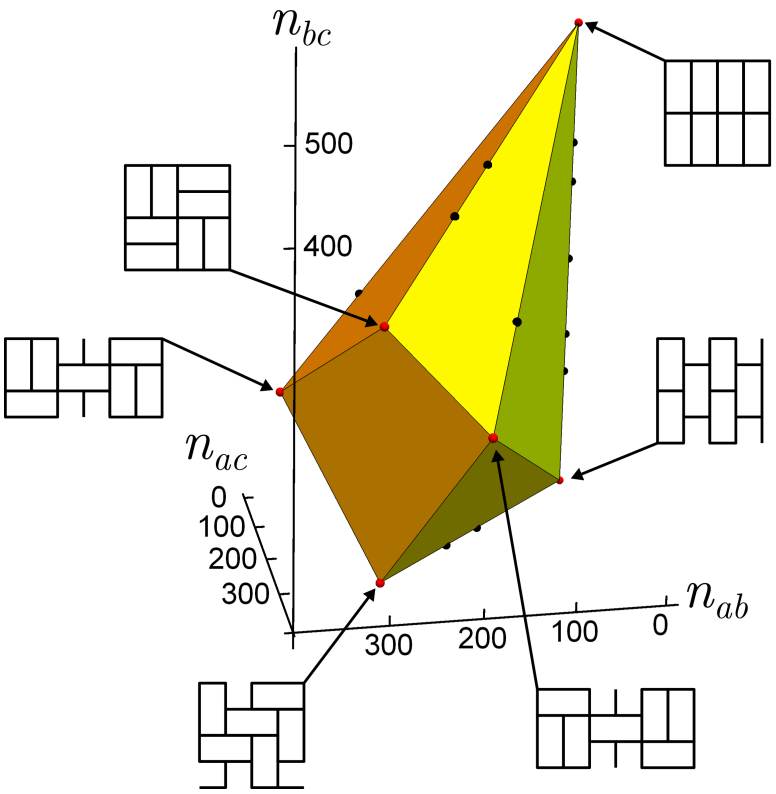


Figure 5.12: The convex hull in $\vec{n}$-space corresponding to all domino patterns with primitive unit cell area up to 8. The vertex configurations are indicated. We also call this convex hull the limit hull, since any convex hull corresponding to all domino patterns with primitive unit cell area up to $A > 8$ must have the same shape as this convex hull. The values of $\vec{n}$ are those that the domino patterns would have if placed within a 24×24 unit cell.

5.5.2 Convex Hull Limit Shape

Enumerating domino patterns with primitive unit cell area up to area A results in sets of domino patterns for which we can take the convex hull, like those shown in Figure 5.11. For different values of A, the convex hull is different. But, at $A = 8$ and greater, the shape of the convex hull stops changing. The boundary of this convex hull constitutes a boundary that cannot be exceeded by any domino pattern (when properly comparing $\vec{n}$-vectors in the way described previously). Consequently, the set of domino patterns with primitive unit cell area up to a value greater than 8 has a convex hull in $\vec{n}$-space of exactly the same shape. This convex hull shape can be seen in Figure 5.11(d), but is shown again in Figure 5.12 with vertex patterns indicated. We will refer to its shape as the limit hull.

The boundary of the limit hull has 7 faces. Each of these faces represents an affine combination of interaction counts that cannot be exceeded. To show this is true, we must make use of all 6 interaction counts $(n_{aa}, n_{ab}, n_{ac}, n_{bb}, n_{bc}, n_{cc})$. Of the 7 faces, 3 lie in the planes $n_{aa} = 0$, $n_{bb} = 0$, and $n_{cc} = 0$. These 3 faces are maximal and cannot be exceeded because none of the interaction counts can be negative. Another face lies in the plane $n_{ac} = 2n_{ab}$. A domino configuration outside this plane would require $n_{ac} > 2n_{ab}$. However, this is not possible because each *a*-*c* interaction necessitates a nearby *a*-*b* interaction. Each *a*-*b* interaction can be placed between two *a*-*c* interactions so that n_{ac} can be equal to $2n_{ab}$ at most. This is shown in Figure 5.13. The face of the convex hull lying in the plane $n_{ab} = 2n_{ac}$ is maximal for the same reason, due to symmetry.

Finally, we have the face lying on the plane $2n_{bb} = NM - n_{ac}$ (where $N \times M$ is the unit cell the configuration is generated by). In this case, we can see that $2n_{bb} > NM - n_{ac}$ is not possible, since each *a*-*c* interaction reduces the number of possible *b*-*b* interactions by 1. The maximum number of *b*-*b* interactions is half of NM; therefore, $2n_{bb} = NM - n_{ac}$ is the maximum possible value for n_{bb} for a given value of n_{ac}. The face lying on the plane $2n_{cc} = NM - n_{ab}$ also cannot be exceeded, due to symmetry. The preceding arguments associate all 7 faces of the limit hull with planes that cannot have domino configurations outside of them.

The 6 vertex patterns in Figure 5.12 are unique to their interaction count vectors, even when considering all possible domino configurations. This means that if a set of domino configurations includes one of these special configurations, there must always exist some interaction parameter vector $\vec{\varepsilon}$ such that the special configuration has the lowest energy. Each of the 6 vertex configurations lies at an intersection of the previously mentioned faces of the boundary of the limit hull. Using the corresponding equations for the bounding planes, it is possible to check that the configurations are unique to their interaction count vector by attempting to build a domino configuration by hand while also keeping to the constraints of the bounding planes. For example, the p4g configuration shown in the upper-left of Figure 5.12 is at the intersection of planes $n_{aa} = 0$, $n_{bb} = 0$, and $n_{cc} = 0$. Starting with a single domino and keeping to these constraints, there is only one possible planar domino pattern. A similar process can be done to show each of the 6 vertex configurations are unique to their interaction count vectors.

For a single unit cell to generate a set of configurations with a convex hull that matches the limit hull, it is necessary and sufficient for that unit cell to generate the 6 configurations shown in Figure 5.12. The 6 vertex configurations can be made by the 6×3 unit cell and 4×4 unit cell. Therefore, any unit cell that is a multiple

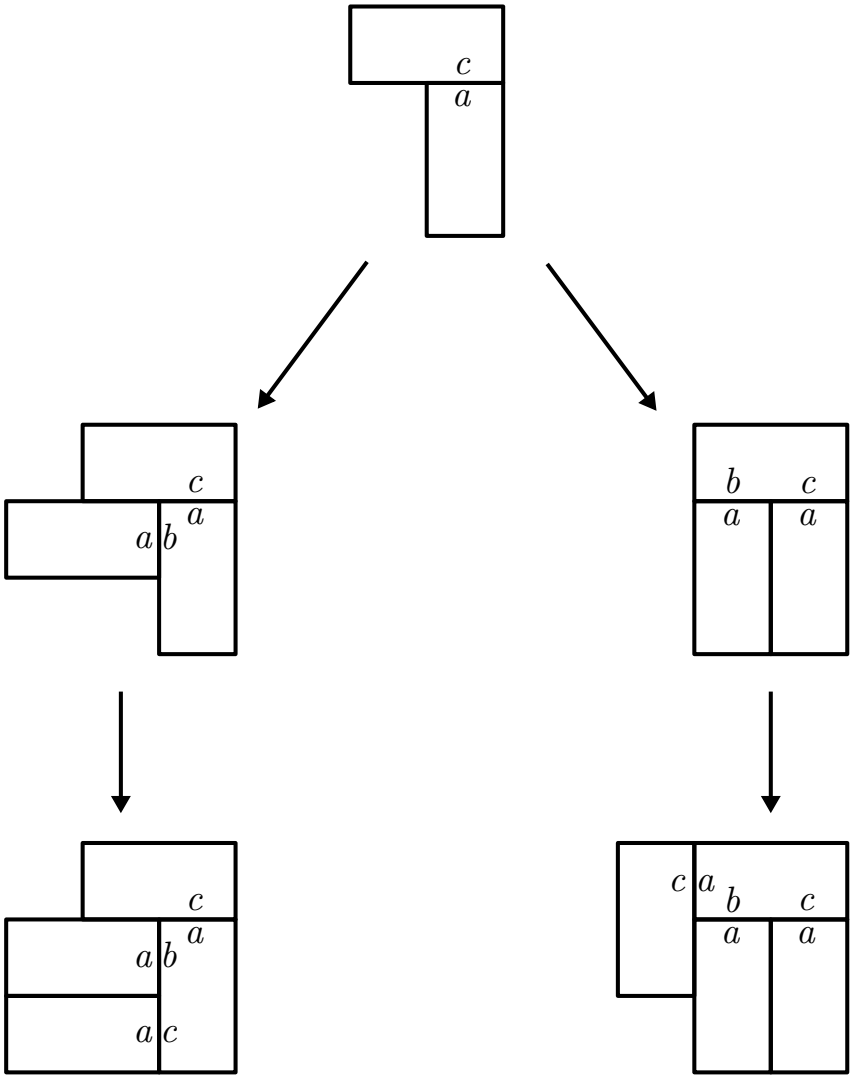


Figure 5.13: Starting with an a-c interaction, a corner is created that can be covered in one of two ways. In either case, an a-b interaction is introduced. From there, the a-b interaction can be shared with at most 1 nearby a-c interaction. Because of this effective constraint, $n_{ac} = 2n_{ab}$ is the maximum possible value for n_{ac}, given the value of n_{ab}.

of the 12×12 unit cell will generate configurations that have the limit hull as the shape of the convex hull.

The concept of a limit hull is useful because it defines the convex hull over the set of all domino patterns. So, considering the set of all planar domino patterns, there are 6 that can be assigned the unique lowest energy. If the periodicity of the domino patterns is constrained, there is a difference between the shape of the convex hull and the limit hull due to the inability to create the 6 vertex patterns of the limit hull. Thinking more generally, it seems possible that similar limit hulls exist for the other polyomino shapes; however, for polyominoes with a greater number of interaction types, it becomes more difficult to identify the convex hull limiting shape as we have done for the domino patterns.

5.6 Domino Flip Connectivity

One can introduce kinetics into the model via the domino flip moves. The subsets of domino configurations that are connected by domino flip moves can be viewed as being kinetically accessible to each other. The notion of kinetic accessibility is introduced in Section 2.3, where we show that this concept is relevant for periodic planar domino patterns, not just for the unit cell regions that generate those planar patterns.

The domino configurations that are connected by a sequence of domino flip moves are in the same homology class. Using the terminology introduced in Section 3.6, the sector of a domino configuration uniquely identifies the homology class it belongs to (excluding ladder configurations). Figure 5.14 shows the domino configurations of the 4×4 unit cell in the sector $(1, 0)$, which are connected by domino flip moves.

The domino configurations in a given homology class are a subset of the full set of domino configurations being considered. Figure 5.15 shows the subset corresponding to the $(1, 0)$ sector of the 4×4 unit cell, plotted in $\vec{n}$-space, giving a new convex hull and new vertex configurations. One vertex of the new convex hull is indicated along with the corresponding domino configuration in Figure 5.15. However, the indicated point in $\vec{n}$-space is not a vertex of the convex hull of the full set of configurations on the 4×4 unit cell. This means that although the indicated domino configuration cannot be assigned lowest energy out of all configurations on the 4×4 unit cell, it may be assigned lowest energy out of the domino configurations in the sector $(1, 0)$. Thus, the kinetic constraint generally introduces new configurations of effective lowest energy. A molecular network that can only rearrange

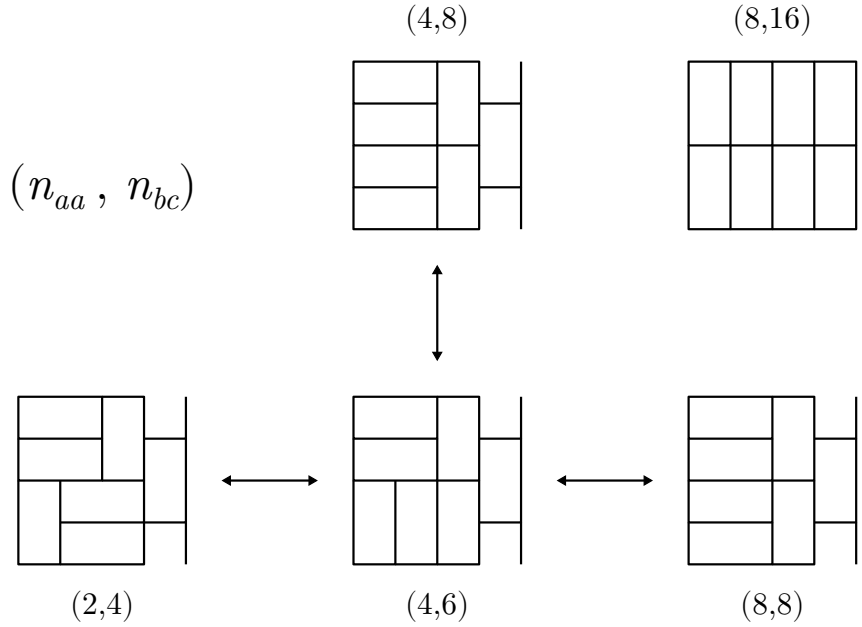


Figure 5.14: Domino configurations on the 4×4 unit cell with labels showing their interaction count vectors. The configurations of the $(1, 0)$ sector are connected by domino flip moves, signified by arrows. The domino configuration in the top-right gives an example configuration in the $(0, 0)$ sector, which is not connected by domino flips to the others because of the difference of the sector.

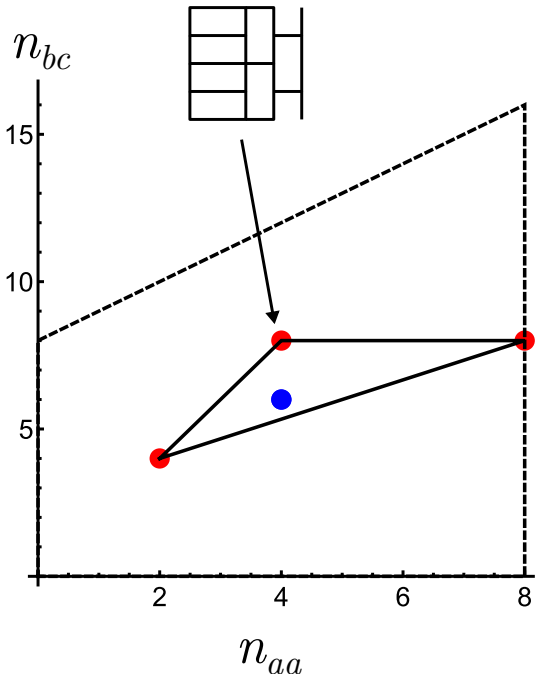


Figure 5.15: Points in $\vec{n}$-space corresponding to domino configurations in the $(1,0)$ sector of the 4×4 unit cell. The convex hull boundary is given by solid lines, vertices are shown as red points, and interior points are shown in blue. One vertex configuration is indicated. The convex hull boundary of the full set of domino configurations of the 4×4 unit cell is shown by the dashed lines.

itself via local moves cannot reach the thermodynamic ground state and is limited to achieving only the lowest energy state consistent with its initial homology class.

5.6.1 Sector Transformations

When applying a Euclidean transformation to a domino configuration, the sector (S_x, S_y) also changes value. Therefore, we find a link between the symmetries of a given domino pattern and the kinetic freedom of that pattern under local rearrangements. The Euclidean transformations that we consider are of the form

$$\begin{pmatrix} x \\ y \end{pmatrix} \to \begin{pmatrix} 0 & -1 \\ 1 & 0 \end{pmatrix}^m \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}^n \begin{pmatrix} x \\ y \end{pmatrix} + \begin{pmatrix} a \\ b \end{pmatrix}. \tag{5.2}$$

Applying this transformation to the domino configuration causes the sector to transform according to

$$\begin{pmatrix} S_x \\ S_y \end{pmatrix} \to \begin{pmatrix} -1 & 0 \\ 0 & -1 \end{pmatrix}^{a+b} \begin{pmatrix} 0 & 1 \\ -1 & 0 \end{pmatrix}^{m} \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}^{n} \begin{pmatrix} S_x \\ S_y \end{pmatrix}, \tag{5.3}$$

which can be calculated by inspection of a few test examples. Some examples of the sector-changing transformations can be seen in Figure 5.16. It is not surprising that rotations, reflections, and glide reflections cause the sector to be transformed. However, a pure translation can also cause the sector to change. This is because the black and white checkerboard used for calculating the sector breaks the translational symmetry of the underlying grid. Translating a domino configuration by (a, b) with $a + b$ odd effectively switches black and white squares, reversing the sector.

Figure 5.17 contains example diagrams for sector transformations, which can be calculated by using Equation (5.2) and Equation (5.3). The top-left diagram of Figure 5.17 shows how the sector is transformed when $S_x \neq S_y$ and both are non-zero. In this case, a reflected domino configuration is in a different sector to any of the sectors of rotated domino configurations. Since the reflected domino configuration cannot be related to the original configuration by any translation or rotation, the domino configuration is chiral. Because of this, the sectors with $S_x \neq S_y$ and both values non-zero contain only chiral domino configurations.

The previous statement made use of the sector of a given domino configuration to give some detail on the possible Euclidean symmetries of that configuration. We can use a similar line of reasoning to give other relations between the sector and symmetries. Suppose a Euclidean symmetry preserves a domino configuration. This implies that the corresponding transformation of the sector also preserves the value of the sector for that domino configuration. Using Equation (5.3), the parameters $n = 0$ and $(m+2(a+b))\mathrm{mod}(4) \neq 0$ correspond to a transformation that rotates the sector. The only sector preserved by a rotation is the zero sector $(0, 0)$; therefore, any domino configuration that is preserved under a Euclidean transformation with these parameters must be in the zero sector.

There are 3 possible types of transformation of the domino configuration that rotate the sector: Firstly, $m = 1$, the transformation is a 4-fold rotation. Secondly, $m = 0$ and $a + b$ odd is a translation that effectively switches the light and dark squares of the checkerboard. Thirdly, $m = 2$ and $a + b$ even is a 2-fold rotation around certain positions. Since $a + b$ is even, these integers can be both even or both odd. However, both odd is not possible because it corresponds to rotation around the middle of a plaquette, which cannot preserve any domino configuration. Therefore, a and b are both even, and the third possibility is equivalent to a 2-fold rotation around an integer point. This third condition is actually a generalisation

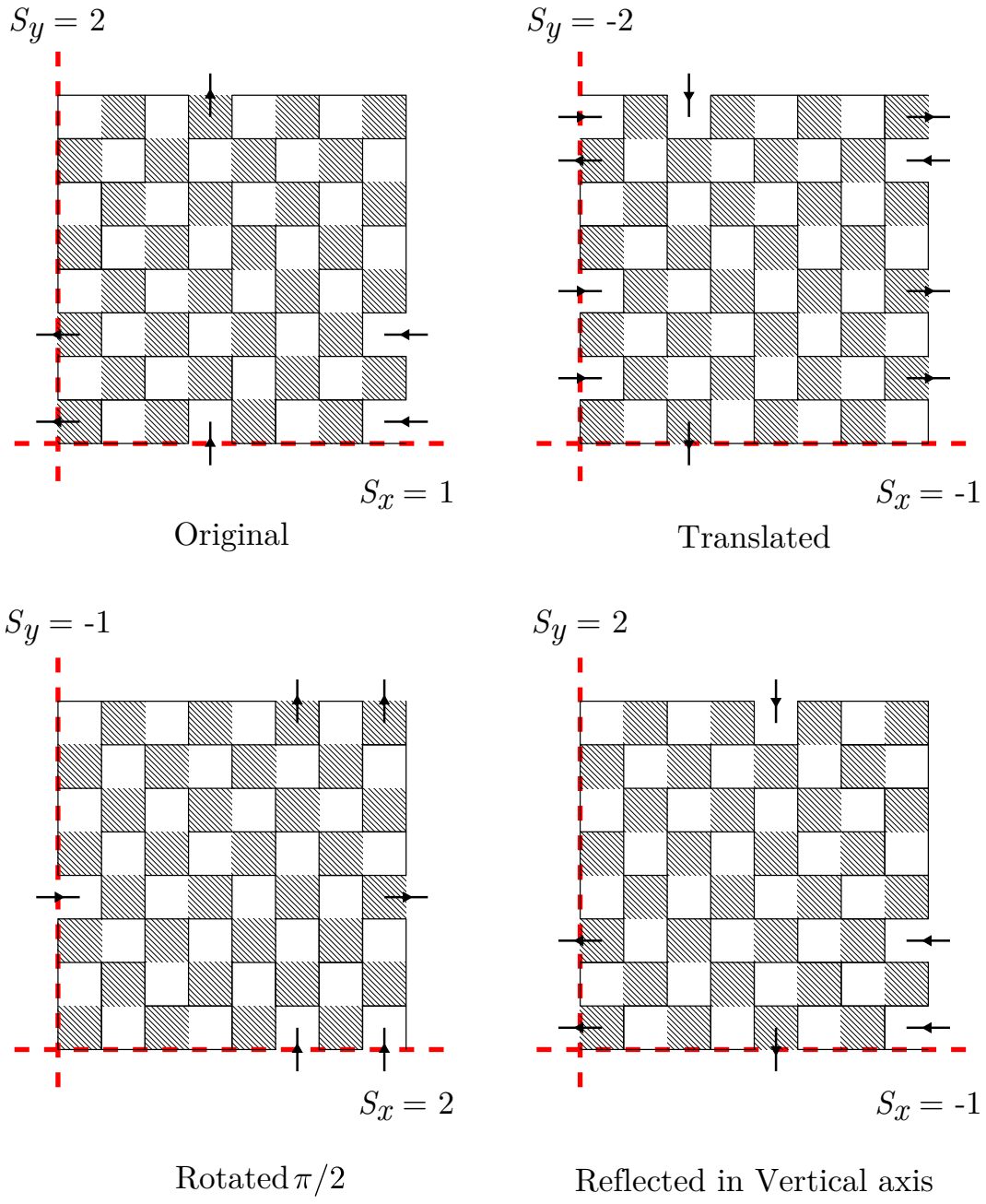


Figure 5.16: A domino configuration on the 8×8 unit cell is shown, with 3 configurations that have been formed by taking Euclidean transformations of the original. In each case, the sector is calculated, and agrees with the values prescribed by Equation (5.3). The translation shown is by 1 grid square to the left, and the rotation and reflection are around the centre of the square unit cell.

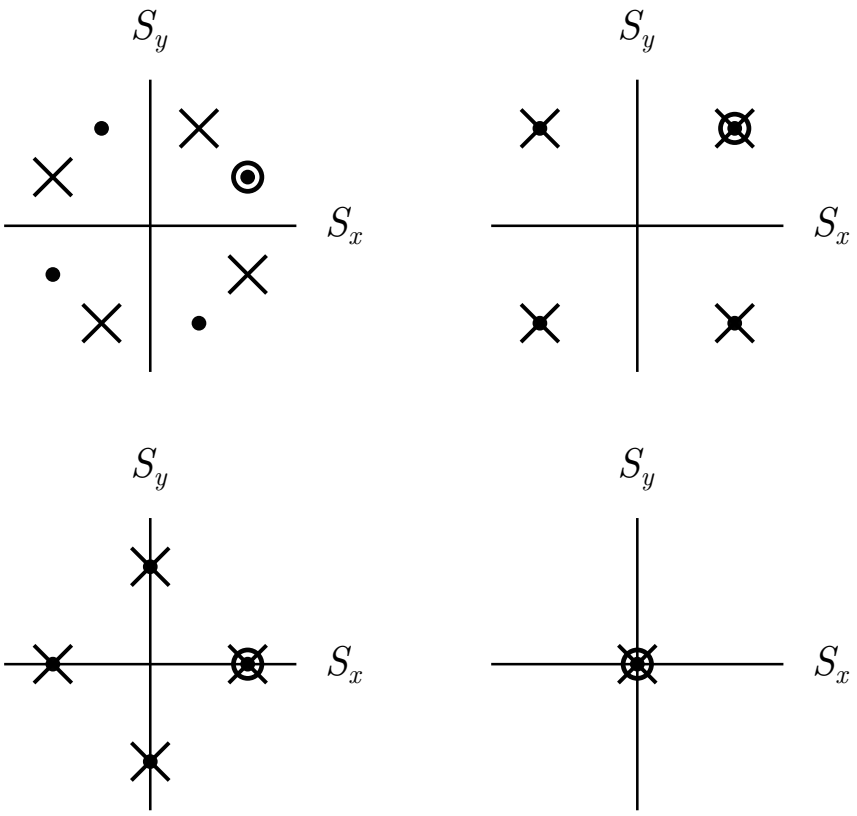


Figure 5.17: For each diagram, consider a domino configuration with sector (S_x, S_y) equal to the position of the circle. Applying reflection to the domino configuration transforms the sector to the position of one of the crosses. Applying rotation or translation to the domino configuration transforms the sector to one of the dots. The dots reside at rotated positions of the original sector. Note there is always a dot at the original sector because there are some rotations and translations of the domino configuration that don't change the sector, for any sector value.

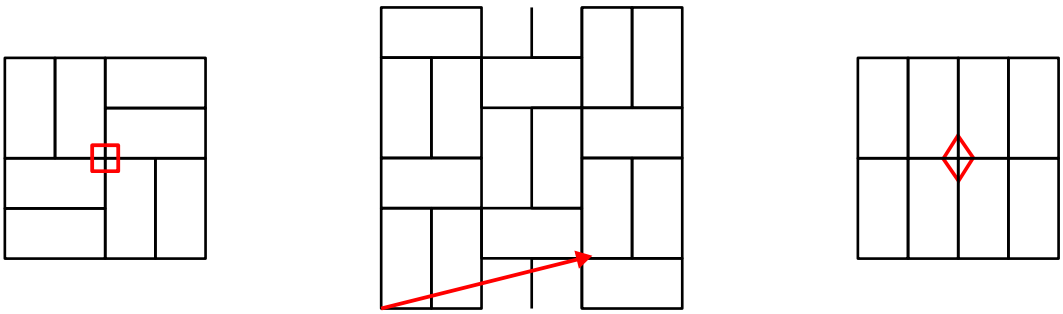


Figure 5.18: 3 Example domino configurations are shown. In each case, their symmetry can be used to infer that they belong to the $(0, 0)$ sector. From left to right, they have: 4-fold rotational symmetry, translation symmetry (a, b) with $a + b$ odd, and 2-fold rotational symmetry around an integer point. The symmetry operations are shown in red.

of the first condition, since any 4-fold rotation must be around an integer position and must imply a 2-fold rotation at the same point. Any domino configuration that is preserved under one of these 3 types of symmetry operation must be in the zero sector. The first case is perhaps most interesting, since it implies that all domino configurations with 4-fold rotational symmetry are contained in the $(0, 0)$ sector. If a system of molecular dominoes falls into a non-zero sector when the molecules become adsorbed to the substrate, there is a kinetic obstruction to being able to form a 4-fold symmetric pattern via local rearrangements.

5.6.2 Partition By Connectivity

The set of domino configurations generated by a unit cell can be partitioned according to their connectivity by domino flip moves, i.e. by their sector. When molecular rearrangements are possible, these subsets of connected configurations determine the kinetic freedom of molecular tilings. However, the transformation of the sector that results from Euclidean transformations of the domino configurations brings up a slight complication. We consider domino configurations that are related by rotations or translations to be equivalent as planar patterns, but the sectors of two domino configurations that we consider equivalent are not necessarily equal.

Looking at Equation (5.2) and Equation (5.3) above, any rotation or translation equivalent domino configuration will be in a rotated sector. Furthermore, any given domino configuration has an equivalent domino configuration in each of the rotated sectors. Therefore, for the subsequent analysis of the connectivity of domino configurations, we include all domino configurations from rotated (i.e. equivalent) sectors into one sector. For example, the $(1, 1)$ sector includes all domino config-

urations from the $(1,1)$, $(1,-1)$, $(-1,1)$, and $(-1,1)$ sectors. By doing this, we coarsen the connectivity partition of domino configurations by placing rotation or translation equivalent configurations into the same subset of our new connectivity partition.

For the 8×8 unit cell there are 7 inequivalent sectors, excluding the sectors of maximum L^1-norm. Each point in $\vec{n}$-space generally corresponds to configurations from multiple sectors. For each sector, Figure 5.19 highlights in red the points in $\vec{n}$-space that correspond to at least one configuration from that sector. The sets of configurations from the various sectors are somewhat localised in $\vec{n}$-space. In particular, the configurations in the greater-valued sectors tend to have a lower value of n_{bc}. This is not surprising, since flow across a cut is caused by rows of dominoes with alternating shift, meaning n_{bc} is low and there are less dominoes available to be flipped. The $(0,0)$ sector takes up the largest region of $\vec{n}$-space, with 1311 points in $\vec{n}$-space corresponding to 604,747 configurations, out of the full 1551 points and 1,224,518 configurations of the 8×8 unit cell.

Taking a convex hull of each set of connected configurations produces 7 different convex hulls, shown in Figure 5.20. Different lowest energy configurations are possible in each case, since each convex hull has a different shape. There are some themes that can be seen across different sectors. For example, the boundary of the convex hull for the $(1,1)$ sector has a face with many associated points in $\vec{n}$-space and has high degeneracy, similar to the high degeneracy faces of the $(0,0)$ sector. The convex hulls for the $(2,1)$ and $(1,2)$ domino sectors of Figure 5.20 are made up of only chiral domino patterns, as explained in Subsection 5.6.1, and these two subsets are related to each other by mirror symmetry.

For each sector, the restriction of configurations by the kinetic constraint gives rise to different possible lowest energy patterns. If the molecular network was able to rearrange macroscopic numbers of molecules simultaneously, there would be no kinetic restriction, and the realised convex hull would be that corresponding to all domino patterns, regardless of their sector. Also, if the molecular network was not fully-packed, the homology description would no longer be valid, and the system of dominoes would be able to rearrange much more freely.

In Figure 5.19 and Figure 5.20, the sectors with maximum L^1-norm are not shown. This is because they are trivial sectors, containing only ladder configurations, which are all isolated under domino flip moves. The ladder configurations were introduced in Subsection 2.3.3. It is clear that ladder configurations are disconnected under the domino flip moves because each ladder configuration has no domino pairs that could be flipped. All domino configurations on a simply-connected region are

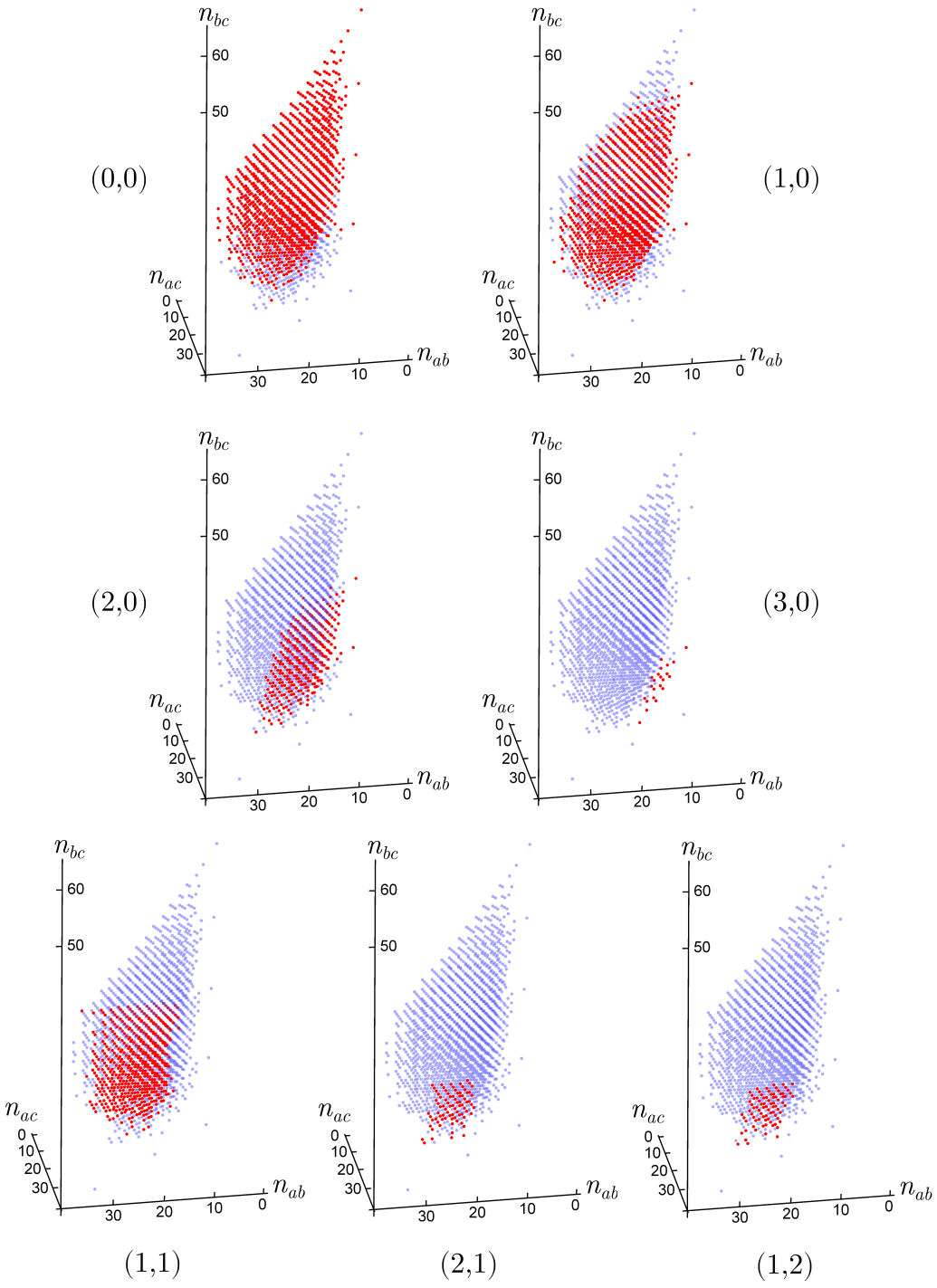


Figure 5.19: The domino configurations of the 8×8 unit cell are represented as points in $\vec{n}$-space. Each plot is labelled by a sector. The points highlighted red in a given plot are the points in $\vec{n}$-space that correspond to at least one domino configuration in the given sector. Other points of $\vec{n}$-space for the domino configurations of the 8×8 unit cell are coloured blue.

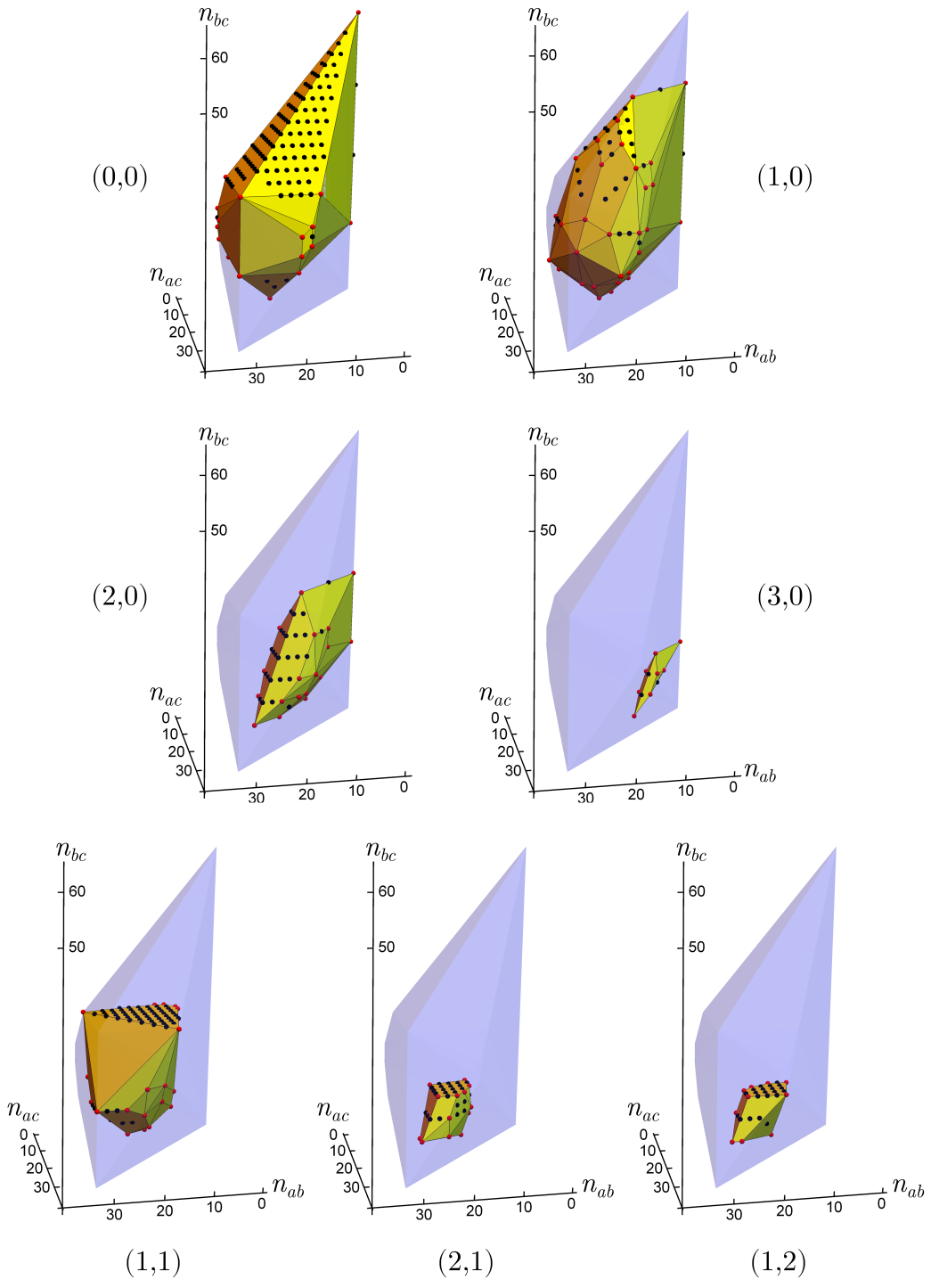


Figure 5.20: The convex hulls for the 7 inequivalent sectors with L^1-norm less than maximum for the domino configurations of the 8×8 unit cell are shown in $\vec{n}$-space with vertex points in red and other boundary points in black. A blue shadow is used to show the convex hull of the set of all domino configurations on the 8×8 unit cell. Each convex hull is labelled by the value of the sector it corresponds to.

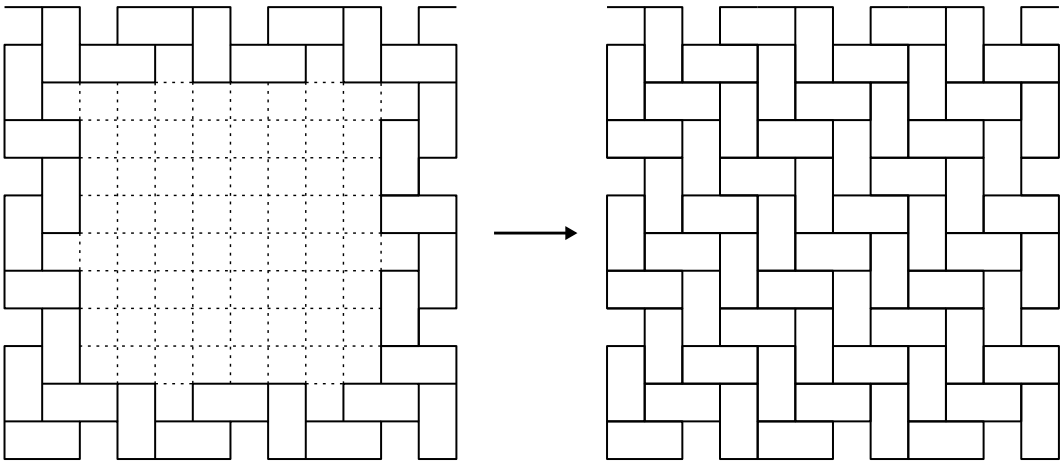


Figure 5.21: On the left, a partially completed domino configuration is shown with underlying grid illustrated by dashed lines to aid the viewer. There is only one way to complete the configuration, which is by using the Herringbone pattern on the right.

connected by domino flip moves [43]. Therefore, thinking of a ladder configuration as tiling the plane, it is not possible to rearrange any finite region of a planar ladder pattern to make another fully-packed domino pattern, no matter how large the region. This is a specific property of the ladder configurations. An example is shown in Figure 5.21, where it can be seen by inspection that a fully-packed domino configuration can only be completed by using the Herringbone pattern. The empty region was formed by taking dominoes out of a Herringbone pattern, which is a ladder configuration. Therefore, the dominoes can only be replaced in the same way.

5.7 Space Of Interaction Counts

We have already considered the possible lowest energy domino patterns, which reside on the boundary of the convex hull as points in $\vec{n}$-space. However, considering patterns within the convex hull, it is possible to see how properties of domino patterns vary throughout the interaction count space. This gives an indication of how the interaction parameters promote a given property. Figure 5.22 shows several quantities within the space of interaction counts for the domino patterns generated by the 8×8 unit cell. In each case, a given quantity is calculated for each domino pattern and the average of that quantity is taken over patterns with the same $\vec{n}$-value, giving a mean quantity value for that position in $\vec{n}$-space. Plots (a), (b), (c), and (d) of Figure 5.22 correspond to quantities of: degeneracy, fraction of chiral

patterns, fraction of p1 patterns, and L^1-norm of the sector, respectively.

In Figure 5.22(a), the degeneracy can be seen to take greater values towards the middle and upper part of the convex hull in $\vec{n}$-space. This indicates that while most domino patterns cannot be assigned lowest energy, the boundary components with the highest degeneracy are the two faces on the upper side of the boundary of the convex hull. These are the faces $NM = n_{ab} + n_{bc}$ and $NM = n_{ac} + n_{bc}$ described in Section 5.4. The values in $\vec{n}$-space of the fraction of p1 domino patterns, shown by Figure 5.22(c), increase more steeply towards the centre of the convex hull than the degeneracy. The p1 patterns have lowest symmetry, so most of the more symmetric domino patterns have $\vec{n}$-vectors located on, or close to, the boundary of the convex hull. This is to be expected because the domino patterns at the boundary of the convex hull have extremal interaction counts. To achieve extremal interaction counts, a high degree of symmetry is often required. For example, the domino patterns with highest value of n_{aa} must have orientational symmetry and align in rows.

Figure 5.22(b) shows the proportion of chiral domino patterns. As discussed in Section 5.2, the achiral wallpaper groups pm, pg, cm, pmm, pmg, pgg, cmm, p4m, and p4g are only possible on the achiral plane $n_{ab} = n_{ac}$, which can be seen passing through the middle of the convex hull. Points in $\vec{n}$ space not on this plane correspond solely to chiral domino patterns. The quantity used for Figure 5.22(d) is the L^1-norm of the sector of the domino patterns. The subsets formed by the different sectors have already been discussed in Subsection 5.6.2, where it is mentioned that the sectors of greater value tend to correspond to lower values of n_{bc}. This can be seen even more clearly here, where the mean values for $\mid S_x \mid + \mid S_y \mid$ appear to be stratified within $\vec{n}$-space.

5.8 Density Of States And Heat Capacity

Choosing a direction for $\vec{\varepsilon}$ defines the relative interaction parameters. This is the first step in calculating system properties that depend on this choice, such as the density of states and heat capacity. These system properties are important for determining the thermodynamic behaviour of an ensemble of domino configurations. In this section, we consider the ensemble of domino configurations generated by the 8×8 unit cell, since this is the largest unit cell that can be enumerated with algorithm DLX in reasonable time.

The domino configurations within hyperplanes in $\vec{n}$-space perpendicular to the vector $\vec{\varepsilon}$ correspond to equal energy domino configurations. By including domino

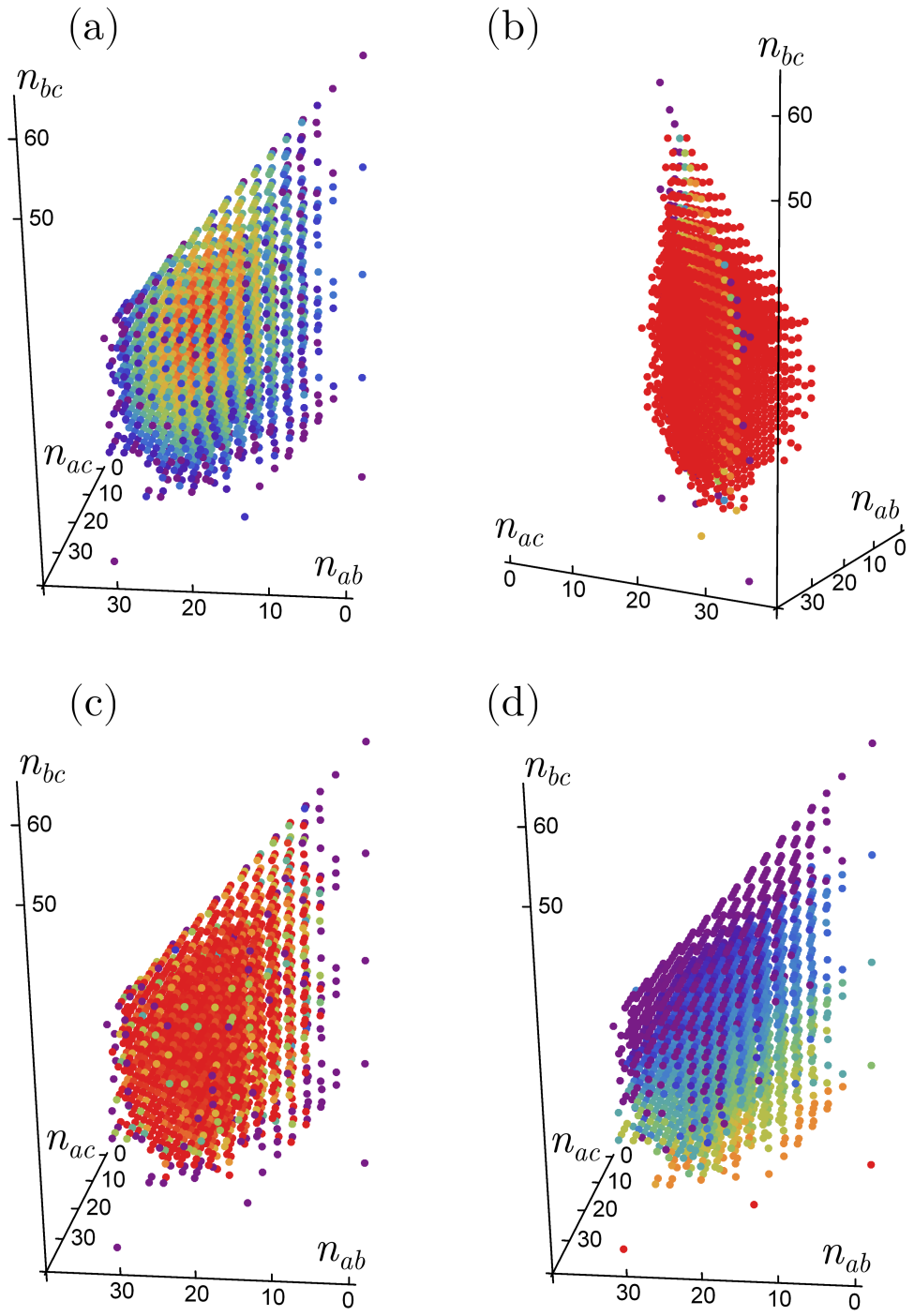


Figure 5.22: Points of interaction space corresponding to at least one domino pattern generated by the 8×8 unit cell are shown as coloured dots. The colour of each dot corresponds to the mean value of a quantity calculated for the domino patterns corresponding to that point in $\vec{n}$-space. The quantities used for plots (a), (b), (c), and (d) are degeneracy, fraction of chiral patterns, fraction of p1 patterns, and L^1-norm of the sector, respectively. The colour scale is logarithmic, with red meaning the greatest value and blue meaning the least.

configurations from nearby hyperplanes for each bin, we obtain a histogram of degeneracy versus energy, which constitutes the density of states. Note that increasing the magnitude of $\vec{\varepsilon}$ increases the energy difference between domino patterns, but does not change the overall shape of the density of states histogram, as long as the bin width is changed proportionately. Therefore, we can use unit magnitude $\vec{\varepsilon}$ for our density of states histogram, without loss of generality.

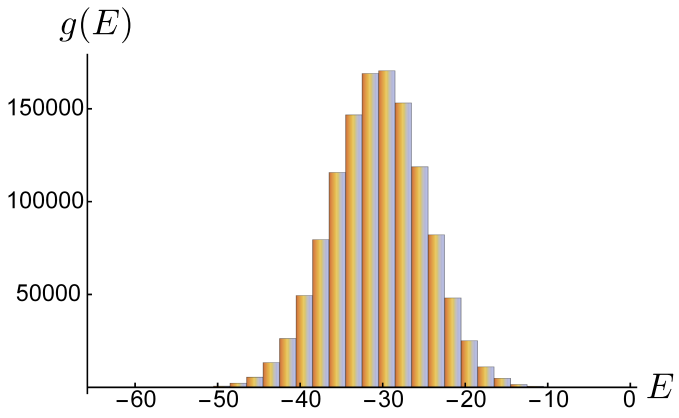
For the domino configurations of the 8×8 unit cell, Figure 5.23 shows the density of states for 3 example choices of $\vec{\varepsilon}$: $(0, 0, -1)$, $(-1, 1, 0)/\sqrt{2}$, and $(-1, -1, -2)/\sqrt{6}$. The overall shape of the density of states depends on the vector $\vec{\varepsilon}$. For example, $(-1, 1, 0)/\sqrt{2}$ is perpendicular to the achiral plane; therefore, the density of states is symmetric around the mid-range value of the energy. Furthermore, $\vec{\varepsilon} = (-1, 1, 0)/\sqrt{2}$ leads to a smaller range of energy values compared to $\vec{\varepsilon} = (0, 0, -1)$, since the convex hull in $\vec{n}$-space has a smaller diameter in the former direction. The third example vector is $(-1, -1, -2)/\sqrt{6}$, which gives a density of states that has a peak close to the lowest energy value $E = -128/\sqrt{6}$, as can be seen in Figure5.23. In this case, the lowest energy state corresponds to 2276 domino configurations, on an edge of the convex hull $(n_{ab}, n_{ac}, n_{bc}) = (32, 32, 32) + \gamma(-1, -1, 1)$ with $\gamma \in [0, 32]$. The comparatively high degeneracy of the lowest energy state can be explained by the proximity of the peak of the density of states.

The density of states can be used to calculate system quantities that depend on the temperature, such as the energy, heat capacity, and the probability assigned to a given energy level. Until now, we have used interaction parameters $\vec{\varepsilon}$ that are defined relative to the temperature. To consider the temperature as a variable while keeping the true interaction parameters fixed, we can use $\vec{\varepsilon}|_{\beta=1}$ to represent the true interaction parameters, so $\vec{\varepsilon} = \beta\ \vec{\varepsilon}|_{\beta=1}$ gives us the explicit dependence on the inverse temperature β. Note that when the magnitude of $\vec{\varepsilon}|_{\beta=1}$ is greater, the interactions are overall stronger, which means the transition between order and disorder occurs at a lesser value of β. Apart from this effect, the behaviour of the system can be understood by choosing $\vec{\varepsilon}|_{\beta=1}$ to be of unit magnitude. Furthermore, this choice gives the simple equation $|\ \vec{\varepsilon}\ | = \beta$.

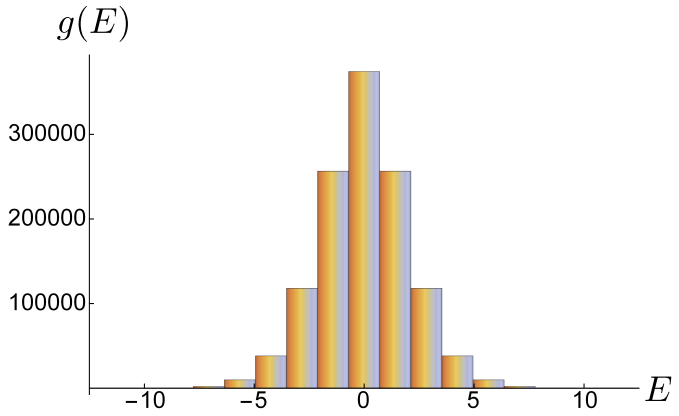
With this in mind, we denote the weight of the ith point $\vec{n}^i$ in interaction count space as

$$w_i = g_i \exp(-\vec{n}^i \cdot \vec{\varepsilon}), \tag{5.4}$$

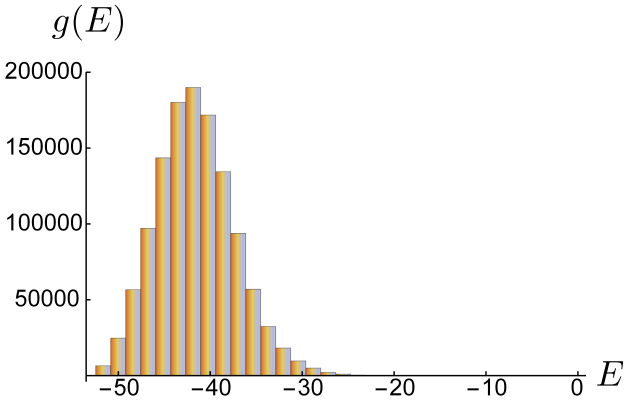
so the probability assigned to the lowest energy state is



$\vec{\varepsilon} = (0, 0, -1)$



$\vec{\varepsilon} = (-1, 1, 0)/\sqrt{2}$



$\vec{\varepsilon} = (-1, -1, -2)/\sqrt{6}$

Figure 5.23: Degeneracy versus energy is shown using 3 example interaction parameter vectors, for the set of domino configurations on the 8×8 unit cell.

$$P_0 = \frac{w_0}{\sum_i w_i}. \tag{5.5}$$

Making use of $\vec{\varepsilon}|_{\beta=1}$ having unit magnitude, we obtain equations for the energy

$$E = \frac{\sum_i (\vec{n}^i \cdot \vec{\varepsilon}) w_i}{\mid \vec{\varepsilon} \mid \sum_i w_i} \tag{5.6}$$

and the heat capacity

$$C = \frac{\sum_i (\vec{n}^i \cdot \vec{\varepsilon} - \mid \vec{\varepsilon} \mid E)^2 w_i}{\sum_i w_i}. \tag{5.7}$$

For the 3 example interaction parameter vectors that were used previously $\vec{\varepsilon}|_{\beta=1}$: $(0,0,-1)$, $(-1,1,0)/\sqrt{2}$, and $(-1,-1,-2)/\sqrt{6}$, we calculate the the probability assigned to the lowest energy state and the heat capacity as functions of $\beta = \mid \vec{\varepsilon} \mid$ for the set of domino configurations of the 8×8 unit cell. The curves for these functions are shown in Figure 5.24. The curve P_0 against β describes the transition from the high entropy state at low β to the lowest energy state at high β. We define β_c to be the inverse temperature at which $P_0 = 0.5$. This is the crossover point between the high entropy state and lowest energy state, where the lowest energy state is assigned half the probability.

The transition for the interaction parameter vectors $(-1,1,0)/\sqrt{2}$, and $(-1,-1,-2)/\sqrt{6}$ occurs at values $\beta_c = 1.7968$ and $\beta_c = 1.6211$. These values are much higher than the transition $\beta_c = 0.61403$ for the interaction parameter vector $(0,0,-1)$ because the density of states of both the former interaction parameter vectors are relatively close to the lowest energy states, shown in Figure5.23. The heat capacity peak also gives an indication of the transition, which can be seen in Figure 5.24. However, the heat capacity curve gives us other information. For the example parameters $\vec{\varepsilon}|_{\beta=1} = (-1,-1,-2)/\sqrt{6}$, the heat capacity curve is less strongly peaked than for the other examples because the density of states is peaked close to the lowest energy state, giving it a high degeneracy, as was shown in Figure 5.23.

5.8.1 Heat Capacity With Multiple Parameters

To get a wider picture of the heat capacity curves for many different directions of $\vec{\varepsilon}|_{\beta=1}$, we can plot the heat capacity as a function of $\vec{\varepsilon}$. Again, we are analysing the set of domino configurations on the 8×8 unit cell. To make 2-dimensional plots, we restrict to $\varepsilon_{ab} = \varepsilon_{ac}$ and $\varepsilon_{bc} = 0$ for plots (b) and (d) of Figure 5.25, respectively. These plots should be thought of in the polar coordinates, since $\vec{\varepsilon} = \beta\, \vec{\varepsilon}|_{\beta=1}$ and we

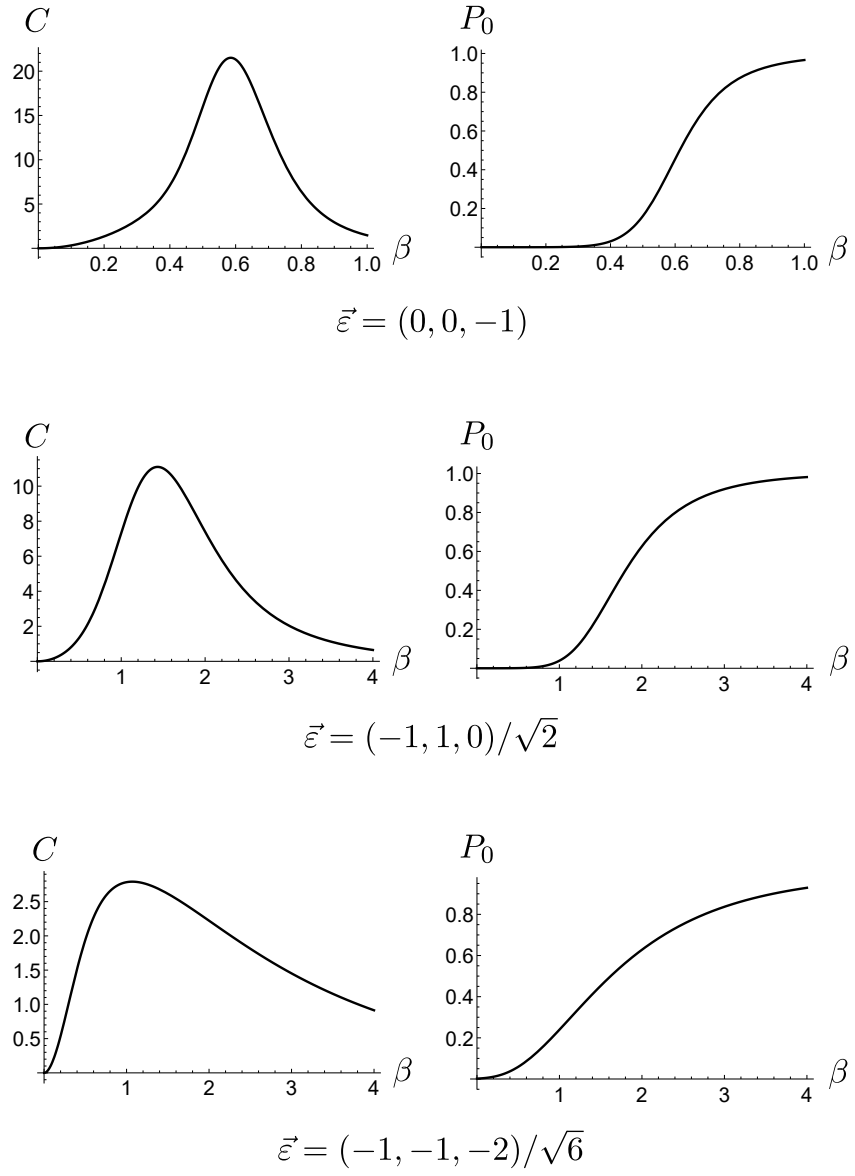


Figure 5.24: The heat capacity C and the probability P_0 assigned to the lowest energy state are shown as functions of inverse temperature β, using 3 example interaction parameter vectors. These 3 vectors are the same as those used in Figure 5.23.

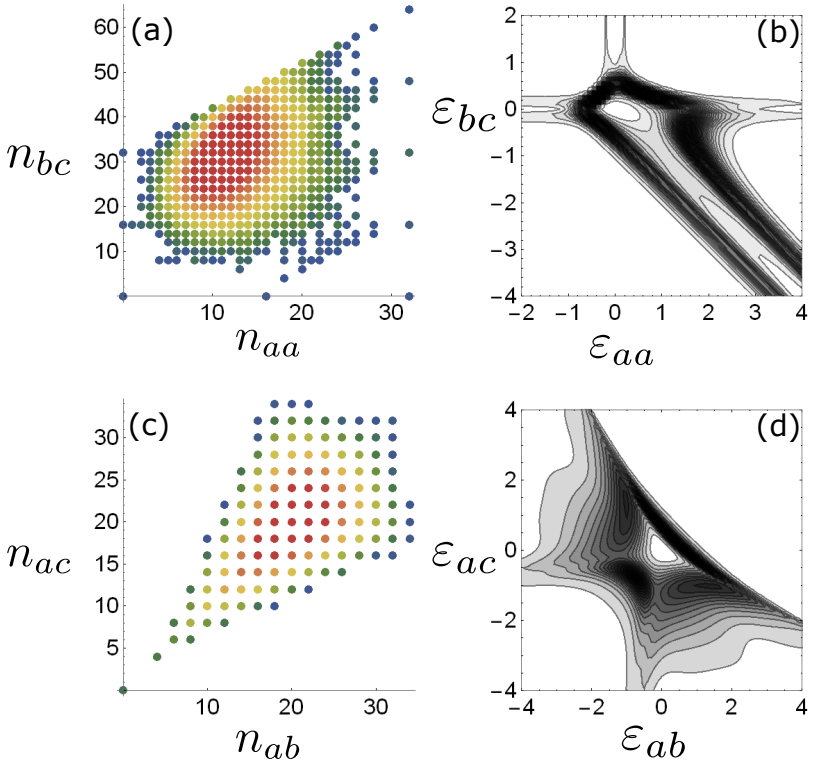


Figure 5.25: The set of configurations considered here are the domino configurations of the 8×8 unit cell. For (a) and (b) we confine $\vec{\varepsilon}$ to the achiral plane, and for (c) and (d) we set $\varepsilon_{bc} = 0$. Shown in (a) and (c) is the degeneracy of configurations; the colour scale goes from low (blue) to high (red), representing the logarithm of degeneracy. (b) and (d) give heat capacity contour maps against $\vec{\varepsilon}$. For a given ratio of interaction parameters, the distance from the origin is the inverse temperature β.

are considering $\vec{\varepsilon}|_{\beta=1}$ to be of unit magnitude. To regain the usual heat capacity curves of Figure 5.24 from plots (b) and (d) of Figure 5.25, one can look along a radial direction, taking the distance along this line to be the inverse temperature and the intensity of the plot to be the value of the heat capacity. This technique was used in Ref. [99] for a different problem involving heat capacity curves.

Plots (a) and (c) of Figure 5.25 show the degeneracy in $\vec{n}$-space corresponding to projections onto the planes $n_{ab} = n_{ac}$ and $n_{bc} = 0$. Therefore, they give the relevant degeneracy for plots (c) and (d). In particular, the upper-left boundary of Figure 5.25(a) corresponds to the lowest energy state of $\vec{\varepsilon}|_{\beta=1} = (-1, -1, -2)/\sqrt{6}$, which was shown previously to have high degeneracy. In Figure 5.25(b), the corresponding interaction parameter vector points to the lower-right. In this direction, the heat degeneracy peaks at a lower value and is more spread out than in other

directions.

The four outer regions of Figure 5.25(b) correspond to the four states given by the vertices of the convex hull in Figure 5.25(a). In this sense, the vertices of the convex hull can be assigned lowest energy by a large region of parameter space compared to the edges and faces of the convex hull. The edges and faces of the convex hull have higher degeneracy than the vertices. This means that when an edge or face of the convex hull is the lowest energy state, the heat capacity curve tends to be more spread out, as can be seen in Figure 5.25(b).

The parameter space in plot (d) of Figure 5.25 generally corresponds to chiral interactions, unlike plot (b), which corresponds to the achiral interactions. The upper-left and lower-right regions in Figure 5.25(d) are the left and right handed chiral ground states. The two other predominant regions of the phase diagram are the upper-right region corresponding to all configurations of full orientational order and the lower-left region, which corresponds to the configurations with $n_{aa} = 0$. Each of these regions indicate ground states that could be realised over a range of experimental parameter values.

Keeping the above examples in mind, some general rules become apparent for the design of ground states that are robust to temperature-induced phase transitions. When the vector of interaction parameters has greater magnitude, the interactions are overall stronger, which raises the transition temperature to the high-entropy state. Also important is the direction of the vector of interaction parameters, since this will determine the smoothness of the transition from the ground state to the high-entropy state, as well as influencing the temperature at which this transition occurs. Robustness against transition to the high-entropy state is also strongly affected by the kinetic restrictions we discussed previously. Due to these kinetic restrictions, the ground state configurations with large values of the L^1 norm of the sector are able to access a smaller set of excited states via local rearrangements. In particular, for configurations with extremal L^1 norm of the sector, there is no possibility for rearrangement via domino pair flips.

5.9 Mapping The Space Of Interaction Parameters

We have described how the vertices of the convex hull of points in $\vec{n}$-space correspond to the possible ground states. This means that for a given vertex point $\vec{n}^0$, there exists an interaction parameter vector $\vec{\varepsilon}$ that assigns $\vec{n}^0$ the lowest energy. We can also calculate the range of interaction parameters that give rise to this lowest energy state. Exact control over interactions is often not possible, so it is important

to explore this range. For $\vec{n}^0$ to be strictly lowest energy, we require $E^i > E^0$ for each point in interaction count space $\vec{n}^i$ that corresponds to at least one configuration. Since the energies of configurations can be written as $E^i = \vec{n}^i \cdot \vec{\varepsilon}$, we get a set of equations

$$(\vec{n}^i - \vec{n}^0) \cdot \vec{\varepsilon} > 0, \forall i \neq 0. \tag{5.8}$$

These equations define a system of linear inequalities for $\vec{\varepsilon}$. Considering the point $\vec{n}^0$ as the origin, the set of values of $\vec{\varepsilon}$ that satisfy the above equation is called the dual cone of the set of points $\vec{n}^i$. The dual cone extends outward indefinitely. However, the magnitude of $\vec{\varepsilon}$ does not change the lowest energy state, so by setting $\mid \vec{\varepsilon} \mid = 1$, we find the intersection of the dual cone with the unit sphere, giving the full range of directions of $\vec{\varepsilon}$ that result in the lowest energy state $\vec{n}^0$.

As an example, for the domino configurations on the 8×8 cell, the vertex point $\vec{n} = (0, 0, 64)$ corresponds to the largest proportion of interaction parameter space: 32.2%. This configuration has all dominoes aligned and not shifted with respect to each other. Since this configuration is the lowest energy state under the largest region of parameter space, it would be quite stable under perturbations to the effective interaction parameters. It is also quite stable in the sense that it is in the $\vec{S} = (0, 0)$ sector, which is the trivial homology class, so it can be reached by many other configurations even when considering the kinetic restrictions.

In Figure 5.26, we have illustrated the convex hull for the set of configurations on the 8×8 unit cell, with a sphere surrounding it. Each point on the sphere corresponds to a value $-\vec{\varepsilon}$. Curves on this sphere represent directions of $-\vec{\varepsilon}$ where two vertices of the convex hull have equal lowest energy. We use the negative of the interaction parameter vector because this means vertices of the convex hull are close to their corresponding regions of parameter space, rather than opposite to them.

We can also calculate the fraction of parameter space that corresponds to chiral ground states. For the domino configurations of the 8×8 unit cell, there are 12 vertices corresponding to only chiral configurations. These happen to be the vertices not on the achiral plane. Therefore, the regions of parameter space that select chiral states are those regions of Figure 5.26 that do not intersect the plane $n_{ab} = n_{ac}$. In total, these chiral vertex states cover 13.8% of interaction parameter space. Of these chiral vertices, only four correspond to unique configurations. These non-degenerate chiral ground states only correspond to 3.7% of parameter space. Therefore, designing for a non-degenerate chiral ground state requires precise control of the interaction parameters, which may be more challenging experimentally.

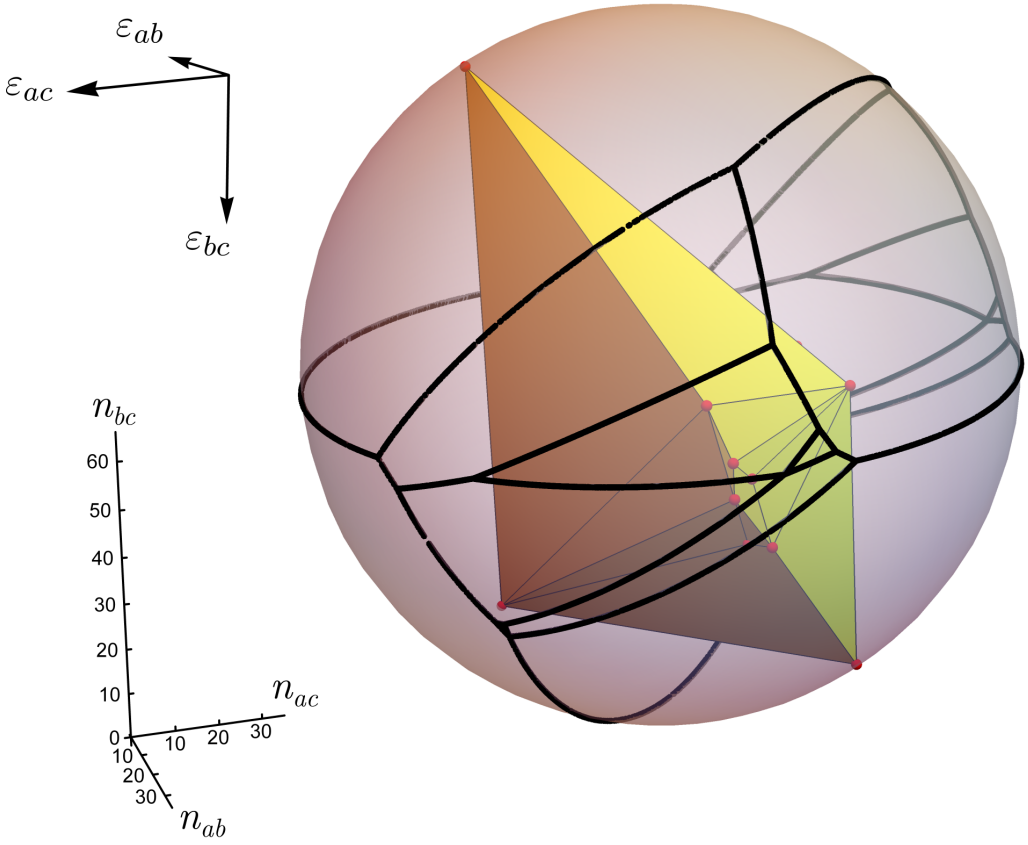


Figure 5.26: The convex hull is shown for the set of domino configurations generated by the 8×8 unit cell with vertices shown as red points. Surrounding the convex hull is a sphere representing the unit magnitude values of $-\vec{\varepsilon}$. The sphere is divided into separate regions, corresponding to the different vertices that are assigned lowest energy by the interaction parameter vector $-\vec{\varepsilon}$. For example, the upper-left region of the sphere corresponds to the values of $-\vec{\varepsilon}$ that select the upper-left vertex as ground state. The reversed axes in the top-left show the coordinates for the interaction parameter vector $\vec{\varepsilon}$.

Chapter 6

Generalisation To Polyominoes

Many of the concepts we have developed for domino configurations have a natural generalisation for configurations of other polyomino types. In particular, the possible energies assigned to a set of domino configurations are determined by the interaction count vector $\vec{n}$ of each configuration and the interaction parameter vector $\vec{\varepsilon}$. For other kinds of polyomino, we have an analogous situation and we can construct convex hulls to identify possible ground states. However, the dimension of the interaction parameter vector $\vec{\varepsilon}$ is different in each case depending on the possible interaction types. In this chapter, we investigate three fully packed interacting polyomino systems: straight trominoes, right trominoes, and T-tetrominoes to illustrate some of the similarities and differences between more general polyomino shapes and the fully packed interacting domino system.

6.1 Wallpaper Groups Of Polyomino Patterns

The symmetry library described in Section 3.3 can be used for general polyomino configurations. Tables 6.2–6.1 show the wallpaper groups generated by various unit cells, for T-tetromino, domino, straight tromino, and right tromino patterns, respectively. In some of the cases shown in the tables, the unit cell does not generate any configurations. For example, the 5×4 unit cell cannot be fully packed with straight trominoes, which is to be expected because the area of this unit cell is not divisible by the area of the straight tromino. Another important point to note is that it can be possible for an $N \times M$ unit cell (i.e. rectangular region with periodic boundary) to be fully packed by a certain polyomino, even when the $N \times M$ rectangle with rigid boundary cannot be fully packed by the same type of polyomino. For example, it has been proven that only rectangles with $4 \mid N$ and $4 \mid M$ can be fully

packed with T-tetrominoes [67], but as can be seen in Table 6.1, the 6×4 unit cell can be fully packed with T-tetrominoes. The number of polyomino configurations corresponding to each wallpaper group are also shown in the tables. Generally, the fraction of configurations with p1 wallpaper group increases with increasing unit cell size. This is because for large unit cells, the majority of configurations have no particular symmetry.

M N	3	4	5	6	7	8	9	10
3								
4		cm(1) pmg(1) pgg(1) cmm(1) p4(2)		p1(4) p2(4) pg(2) cm(1) pmg(2)		p1(18) p2(26) pm(1) pg(7) cm(1) pmg(4) pgg(6) cmm(2) p4(2)		p1(160) p2(32) pg(16) cm(1) pmg(4)
5						p2(4)		
6		p1(4) p2(4) pg(2) cm(1) pmg(2)				p1(48) p2(10) pm(1) pg(9) cm(1) pmg(2) pgg(5)		
7						pg(1)		
8		p1(18) p2(26) pm(1) pg(7) cm(1) pmg(4) pgg(6) cmm(2) p4(2)	p2(4)	p1(48) p2(10) pm(1) pg(9) cm(1) pmg(2) pgg(5)	pg(1)	p1(270) p2(144) pm(1) pg(31) cm(1) pmg(5) pgg(15) cmm(2) p4(8) p4g(1)	p1(64) p2(40) pg(5)	p1(3862) p2(212) pm(1) pg(97) cm(1) pmg(4) pgg(11)
9						p1(64) p2(40) pg(5)		
10		p1(160) p2(32) pg(16) cm(1) pmg(4)				p1(3862) p2(212) pm(1) pg(97) cm(1) pmg(4) pgg(11)		

Table 6.1: The number of T-tetromino configurations of each wallpaper group are shown for unit cells $N \times M$ up to 10×10.

M / N	2	3	4	5	6	7	8
2	pmm(1) cmm(1)	pmm(3)	pmm(3) pmg(1) cmm(2)	pm(1) pmm(7)	pm(2) pmm(9) pmg(4) cmm(3)	pm(9) pmm(17)	pm(19) pmm(23) pmg(11) cmm(6)
3	pmm(3)		pmm(5) pmg(1)		p2(4) pm(2) pmm(8)		p1(6) p2(8) pm(5) pmm(13) pmg(5)
4	pmm(3) pmg(1) cmm(2)	pmm(5) pmg(1)	pm(1) pmm(5) pmg(3) pgg(1) cmm(2) p4g(1)	p1(2) p2(2) pm(7) pmm(15) pmg(6)	p1(6) p2(14) pm(31) pg(1) pmm(31) pmg(18) pgg(1) cmm(7)	p1(52) p2(14) pm(92) pg(1) pmm(49) pmg(25)	p1(218) p2(68) pm(300) pg(6) cm(1) pmm(98) pmg(67) pgg(2) cmm(13) p4g(1)
5	pm(1) pmm(7)		p1(2) p2(2) pm(7) pmm(15) pmg(6)		p1(78) p2(44) pm(53) pmm(31)		p1(1550) p2(204) pm(238) pg(13) pmm(65) pmg(44)
6	pm(2) pmm(9) pmg(4) cmm(3)	p2(4) pm(2) pmm(8)	p1(6) p2(14) pm(31) pg(1) pmm(31) pmg(18) pgg(1) cmm(7)	p1(78) p2(44) pm(53) pmm(31)	p1(360) p2(102) pm(169) pg(9) cm(4) pmm(64) pmg(27) pgg(3) cmm(4)	p1(3716) p2(360) pm(834) pmm(135)	p1(27240) p2(1526) pm(3964) pg(153) cm(12) pmm(513) pmg(227) pgg(22) cmm(31)
7	pm(9) pmm(17)		p1(52) p2(14) pm(92) pg(1) pmm(49) pmg(25)		p1(3716) p2(360) pm(834) pmm(135)		p1(167300) p2(2994) pm(6443) pg(252) pmm(385) pmg(285)
8	pm(19) pmm(23) pmg(11) cmm(6)	p1(6) p2(8) pm(5) pmm(13) pmg(5)	p1(218) p2(68) pm(300) pg(6) cm(1) pmm(98) pmg(67) pgg(2) cmm(13) p4g(1)	p1(1550) p2(204) pm(238) pg(13) pmm(65) pmg(44)	p1(27240) p2(1526) pm(3964) pg(153) cm(12) pmm(513) pmg(227) pgg(22) cmm(31)	p1(167300) p2(2994) pm(6443) pg(252) pmm(385) pmg(285)	p1(1181552) p2(9592) pm(30117) pg(1048) cm(30) pmm(1396) pmg(671) pgg(39) cmm(39) p4(30) p4g(4)

Table 6.2: The number of domino configurations of each wallpaper group are shown for unit cells $N \times M$ up to 8×8. Note that the tables are symmetric under the exchange of the unit cell sides N and M, since the rotated unit cell generates the same polyomino patterns.

M N	3	4	5	6	7	8	9
3	p2(2) pm(1) pmm(1)	p2(2) pm(1) pmm(2) pmg(3)	p2(10) pm(4) pmm(2)	p1(10) p2(14) pm(7) pmm(2) pmg(5)	p1(34) p2(34) pm(14) pmm(3)	p1(122) p2(52) pm(25) pg(1) pmm(4) pmg(12)	p1(396) p2(110) pm(46) pmm(4)
4	p2(2) pm(1) pmm(2) pmg(3)			p2(2) pm(2) pmm(3) pmg(4) cmm(1)			p1(2) p2(6) pm(6) pmm(6) pmg(4)
5	p2(10) pm(4) pmm(2)			p1(2) p2(12) pm(5) pg(1) pmm(3) pmg(2)			p1(18) p2(26) pm(9) pmm(5)
6	p1(10) p2(14) pm(7) pmm(2) pmg(5)	p2(2) pm(2) pmm(3) pmg(4) cmm(1)	p1(2) p2(12) pm(5) pg(1) pmm(3) pmg(2)	p1(20) p2(18) pm(11) pg(3) pmm(3) pmg(7) pgg(1) p4g(1)	p1(86) p2(50) pm(22) pg(6) pmm(6) pmg(6)	p1(324) p2(88) pm(52) pg(14) cm(1) pmm(9) pmg(22) pgg(3) cmm(3)	p1(1288) p2(230) pm(101) pg(31) pmm(14) pmg(17)
7	p1(34) p2(34) pm(14) pmm(3)			p1(86) p2(50) pm(22) pg(6) pmm(6) pmg(6)			p1(700) p2(144) pm(49) pmm(11)
8	p1(122) p2(52) pm(25) pg(1) pmm(4) pmg(12)			p1(324) p2(88) pm(52) pg(14) cm(1) pmm(9) pmg(22) pgg(3) cmm(3)			p1(3056) p2(320) pm(172) pg(32) pmm(23) pmg(32)
9	p1(396) p2(110) pm(46) pmm(4)	p1(2) p2(6) pm(6) pmm(6) pmg(4)	p1(18) p2(26) pm(9) pmm(5)	p1(1288) p2(230) pm(101) pg(31) pmm(14) pmg(17)	p1(700) p2(144) pm(49) pmm(11)	p1(3056) p2(320) pm(172) pg(32) pmm(23) pmg(32)	p1(13612) p2(672) pm(181) pmm(18)

Table 6.3: The number of straight tromino configurations of each wallpaper group are shown for unit cells $N \times M$ up to 9×9.

M / N	3	4	5	6
3	cm(1)	p2(4) pg(3) pmg(1)	p1(2)	p1(10) p2(12) pg(2) cm(1)
4	p2(4) pg(3) pmg(1)			p1(34) p2(24) pg(13) cm(1) pmg(2) pgg(10) cmm(1)
5	p1(2)			p1(194) p2(48) pg(10)
6	p1(10) p2(12) pg(2) cm(1)	p1(34) p2(24) pg(13) cm(1) pmg(2) pgg(10) cmm(1)	p1(194) p2(48) pg(10)	p1(416) p2(86) pg(37) cm(3) pmg(3) pgg(9) p4(6)

Table 6.4: The number of right tromino configurations of each wallpaper group are shown for unit cells $N \times M$ up to 6×6.

6.2 Straight Tromino

The straight tromino has 3 face types and its configurations have a 3-dimensional achiral interaction count vector $\vec{n} = (n_{ab}, n_{ac}, n_{bc})$. A straight tromino system with chiral interactions has a higher-dimensional parameter space, making the analysis more difficult to visualise, and we do not consider them here. The convex hull can be calculated for the straight tromino configurations generated by various unit cells, similarly to the domino system. Figure 6.1 shows the convex hull for the 6×6 and 9×9 unit cells. The 6×6 unit cell generates 64 straight tromino configurations, which have a convex hull of 7 vertices, corresponding to 11 configurations. On the other hand, the 9×9 unit cell generates 14,483 straight tromino configurations, which have a convex hull of 14 vertices made up of 53 vertex configurations.

A striking feature that appears for the 9×9 unit cell but not the 6×6 unit cell are the boundary faces of the convex hull with high degeneracy and many points in $\vec{n}$-space residing on those faces. This feature resembles the highly degenerate faces of the domino configurations generated by the 8×8 unit cell. However, in this case, there are 3 high degeneracy faces, which are highlighted by differently coloured points in Figure 6.2. These 3 faces contain 42 out of the 55 $\vec{n}$-vectors on the boundary of the convex hull and correspond to 205 out of the 623 boundary configurations.

All three of the high degeneracy faces are close to being parallel, such that only 1.28% of interaction parameter space lies in the region between the three highly degenerate faces. This means that each of these faces can be realised as lowest

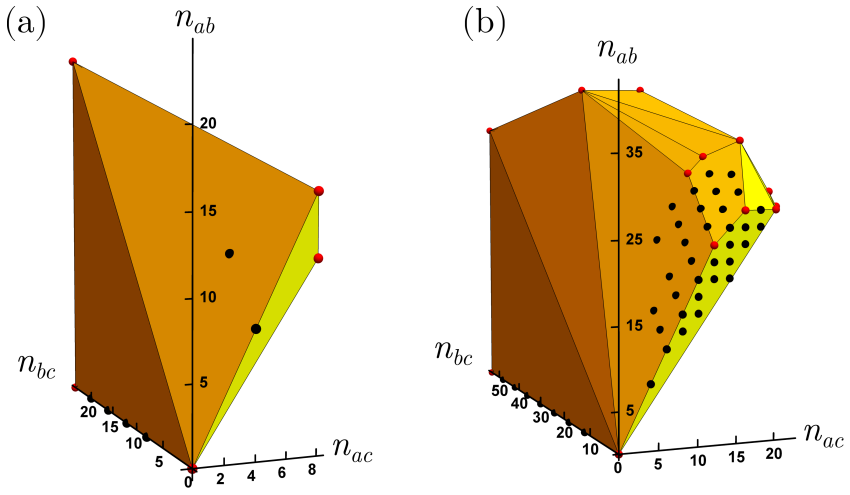


Figure 6.1: The convex hulls of points in $\vec{n}$-space are shown for the straight tromino configurations generated by (a) the 6×6 unit cell and (b) the 9×9 unit cell. Red points are vertices and black points are positions in $\vec{n}$-space corresponding to at least one configuration.

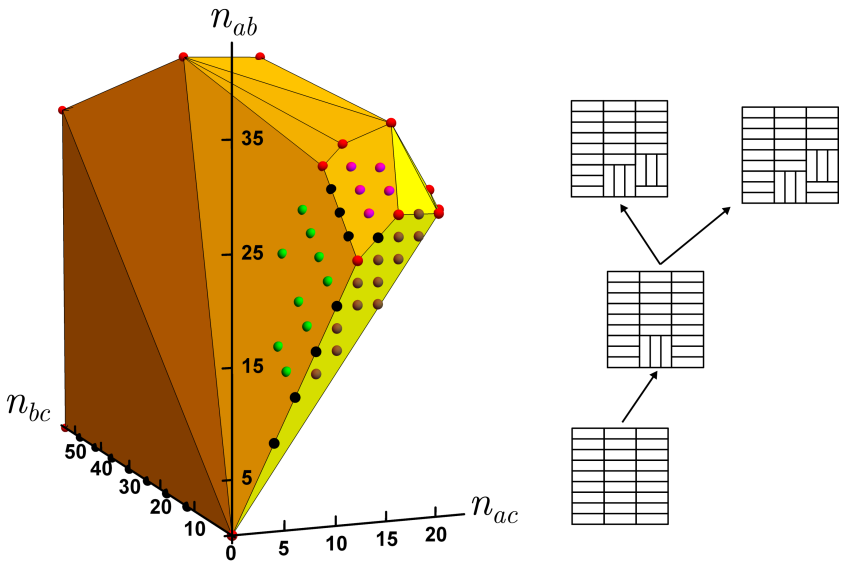


Figure 6.2: The convex hull is shown for straight trominoes generated by the 9×9 unit cell. The 3 faces of high degeneracy are indicated by coloured points in brown, green, and pink. They correspond to the planes $n_{bc} = 0$, $n_{ab} = 2n_{ac} + n_{bc}$, and $n_{ab} = 12 + n_{ac} + n_{bc}$, respectively. On the right, straight tromino flip moves are shown starting from the configuration with $\vec{n} = (0, 0, 0)$ and continuing along the two planes $n_{bc} = 0$ and $n_{ab} = 2n_{ac} + n_{bc}$.

energy state under similar interaction parameters, resulting in sensitive dependence on the form of the interactions between neighbouring straight trominoes. Each of the three high degeneracy faces corresponds to a highly degenerate ground state. These ground states have a relatively high transition temperature to the high-entropy disordered state. This result is similar to that of the high degeneracy ground states of the domino system. As an example, for the straight tromino configurations of the 9×9 unit cell, the vector of interaction parameters $\vec{\varepsilon} = (0, 0, 1)$ corresponds to a highly degenerate ground state with a transition temperature of 1.50625, whereas the parameters $\vec{\varepsilon} = (0, -1, 0)$ correspond to a low degeneracy ground state of transition temperature 0.73366.

Two of the highly degenerate faces just mentioned can be written as straightforward rules, similarly to the highly degenerate faces of the domino system. The face shown by brown points in Figure 6.2 lies in the plane $n_{bc} = 0$, corresponding to configurations where none of the adjacent straight trominoes are shifted by 1 plaquette. The face shown by green points in Figure 6.2 lies in the plane $n_{bb} = 2n_{cc}$, corresponding to configurations with no shifts by 2 plaquettes. Some example configurations are shown on the right of Figure 6.2. These example configurations can be related to each other by local moves involving 3 straight trominoes at once. These moves are the flip moves for straight trominoes described in Chapter 1. By making successive flips as shown in Figure 6.2, it is possible to make local rearrangements while staying on the boundary of the convex hull. The flip shown by the arrow to the left changes (n_{ab}, n_{ac}, n_{bc}) by $(2, 0, 2)$. This breaks the rule $n_{bc} = 0$, but preserves $n_{bb} = 2n_{cc}$. The move to the right changes (n_{ab}, n_{ac}, n_{bc}) by $(2, 2, 0)$, breaking the rule $n_{bb} = 2n_{cc}$ and keeping to the rule $n_{bc} = 0$. Since these faces can be traversed by local moves, we have another analogy to the domino system, where there are high degeneracy lowest energy states that can be traversed by local moves.

To investigate connectivity via local moves for straight trominoes, we compare the periodic boundary conditions to rigid boundary conditions. Under rigid boundary conditions, all straight tromino configurations are connected by flip moves, but this is not the case for periodic boundary conditions. It is important to note that although we do not allow the straight trominoes to cross the boundary, we still count the interactions between straight trominoes on either side of the boundary because we are investigating the issue of connectivity via flip moves, not the issue of physical boundaries. Figure 6.3(a) and (b) show the convex hull for the configurations of the 9×9 unit cell, for periodic boundary and rigid boundary, respectively. One can see from this figure that the rigid boundary only allows configurations with lower values of n_{bc} and the shape of the convex hull becomes more compact. The

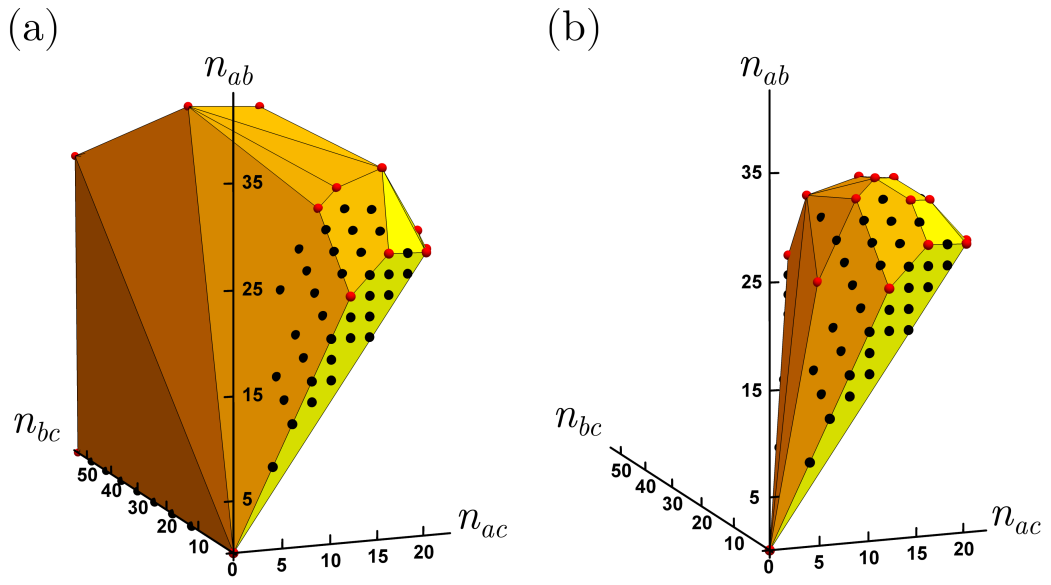


Figure 6.3: The convex hulls of points in $\vec{n}$-space are compared for the periodic boundary (a) and rigid boundary (b), for the straight tromino configurations generated by the 9×9 unit cell.

configurations with high n_{bc} values are only possible by using straight trominoes that are parallel and shifted, but these configurations are more likely to be disconnected via flip moves because they contain many shifts. Similar to the domino system, configurations of shifted straight trominoes seem to be less well connected by flip moves and correlate with a particular direction in $\vec{n}$-space.

Lastly, we can view properties of the straight tromino configurations within the space of interaction counts, as we did for domino configurations. Figure 6.4 shows two quantities within $\vec{n}$-space: the degeneracy and the fraction of chiral patterns. Again, similar to the domino configurations, the degeneracy tends to increase going inwards from the boundary of the convex hull of points. However, in contrast to the domino system, the component n_{bc} is quite different to the other components. There are many points along lines in the n_{bc} direction. This implies this is an easy axis, in terms of generating configurations which differ by only n_{bc}. The n_{bc} interaction occurs when two adjacent straight trominoes are aligned and shifted by 1 plaquette with respect to each other, so this easy axis is enabled by the ability of adjacent straight trominoes to be shifted.

In Figure 6.4(b), the achiral configurations are more evenly spread within $\vec{n}$-space in comparison to the equivalent Figure 5.22(b) for dominoes. This is because we are using straight trominoes with achiral interactions. Indeed, our 3-dimensional

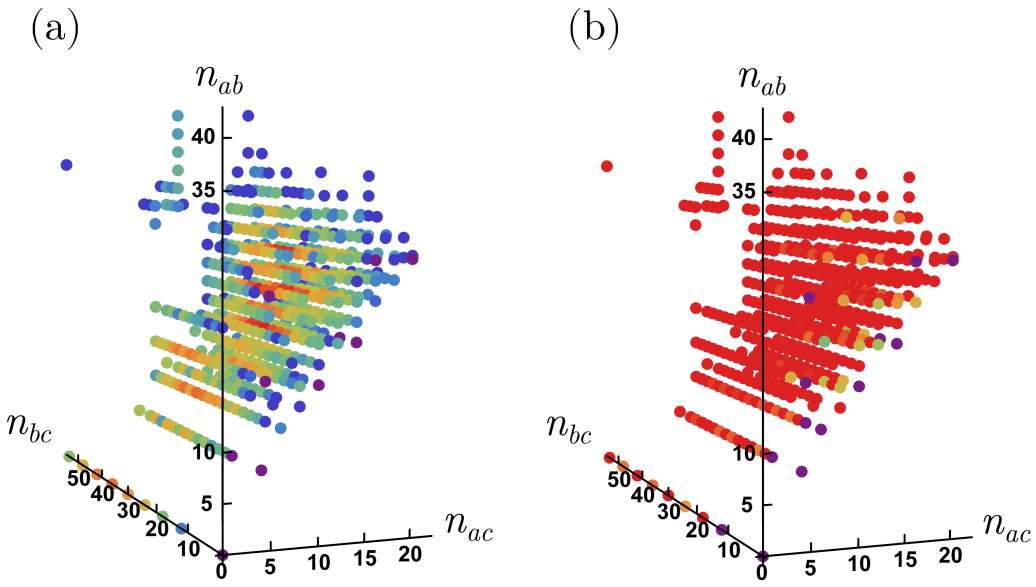


Figure 6.4: Coloured points are shown that correspond to mean quantities calculated for the configurations of equal $\vec{n}$-vector, within the set of straight tromino configurations that are generated by the 9×9 unit cell. The quantities used are: (a) degeneracy and (b) fraction of chiral patterns. The colour scale is logarithmic in the value of the quantity, with red being the greatest value and blue being the least.

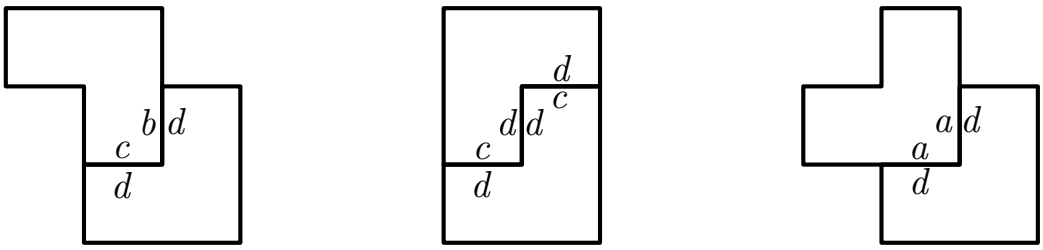
interaction count space for achiral straight trominoes can be imagined as residing within the achiral hyperplane of the space of interaction counts for the chiral straight trominoes. One can also see that in Figure 6.4(b), the points with the lowest values of n_{bc} have a slightly higher proportion of achiral configurations, due to the higher symmetry associated with aligned and unshifted straight tromino configurations.

6.3 Right Tromino

The right tromino has a six-dimensional interaction count vector

$$\vec{n} = (n_{ab}, n_{ac}, n_{bc}, n_{ad}, n_{bd}, n_{cd}). \tag{6.1}$$

This 6-dimensional vector is the reduced set of interaction counts, after making use of the packing constraints due to the fully-packed nature of our configurations. However, for right trominoes, there is an additional packing constraint, which means the right tromino configurations reside in a 5-dimensional hyperplane given by the equation $NM/3 = n_{cd} + n_{ad}/2$ for unit cell with dimensions $N \times M$. This is ex-



1 Corner, 1 b-d, 1 c-d 2 Corners, 2 c-d, 1 d-d 1 Corner, 2 a-d

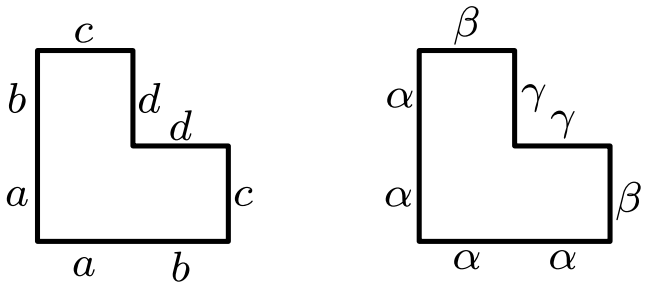
Figure 6.5: The 3 rotation and reflection inequivalent arrangements that take up the space next to a concave corner of a right tromino. Note that in the second case, two such spaces are used up.

emplary of the general behaviour for complicated polyominoes; their configurations generally reside in some subspace of the full possible space of interaction counts.

The additional packing constraint arises from the interactions that occur at the concave corners of the right trominoes. For a given unit cell, there are $NM/3$ right trominoes, so there must also be this many right tromino concave corners. Up to rotations and reflections, there are three ways to arrange right trominoes to occupy the space next to the concave corner, shown in Figure 6.5. Note that the reflected arrangements are identical in terms of interactions, since we are using right trominoes with achiral interactions. For each of the three arrangements, there is either (not both and not neither) of 1 c-d or 2 a-d interactions per concave corner. Therefore, the total number of concave corners of right trominoes in the unit cell is given by $NM/3 = n_{cd} + n_{ad}/2$.

The effective interaction count space for the right tromino is 5-dimensional. Although it is possible to investigate the 5-dimensional convex hull, it is more difficult to visualise. For example, the convex hull of the right tromino configurations generated by the 9×6 unit cell has 114 vertices corresponding to 296 configurations. However, this convex hull cannot be visualised in 3 dimensions. Investigating a subspace of the possible interactions can obtain some insight to the problem. For even more complicated polyominoes, there can be many possible boundary faces for the convex hull, but this is also accompanied by a great increase in the required computational resource, since it is generally difficult to calculate convex hulls for points in many dimensions. For these more complicated polyomino shapes, there may only be experimental control over the strength of some interaction types and not others, so there may still be a possibility to use the convex hull construction to make analysis of the reduced problem.

To demonstrate the use of the convex hull construction on a reduced inter-



$$\begin{pmatrix} 0 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} n_{ab} \\ n_{ac} \\ n_{bc} \\ n_{ad} \\ n_{bd} \\ n_{cd} \end{pmatrix} = \begin{pmatrix} n_{\alpha\beta} \\ n_{\alpha\gamma} \\ n_{\beta\gamma} \end{pmatrix}$$

Figure 6.6: Right tromino interactions of reduced generality are introduced. The matrix equation gives the relation between interaction count vectors of the more general interaction space and the reduced space.

action count space, we define less general right tromino interactions by letting the interaction types on the long edge of the right tromino become equal. In this way, one can imagine that the interaction strength is uniform along the long edge. The new and old interaction definitions and the matrix transformation relating them are shown in Figure 6.6. The matrix equation results from the fact that both a and b face types of the general system correspond to the α face type of the new system. We have not shown the interactions of same face types, which can be eliminated by the packing constraints described in Subsection 3.5.2 in both cases. Also note that despite the additional packing constraint $NM/3 = n_{cd} + n_{ad}/2$, the right tromino configurations span the newly defined 3-dimensional reduced interaction count space.

Having defined the reduced space of interaction counts, we can use the convex hull construction as before. For right tromino configurations generated by the 9×6 unit cell, we obtain the convex hull shown in Figure 6.7. We will call this the reduced convex hull, of the reduced interaction space, in comparison to the full convex hull for the full interaction space. The reduced convex hull has 11 vertices corresponding to 200 configurations. For the reduced interaction space, the number of configurations per vertex point is an order of magnitude greater than for the full

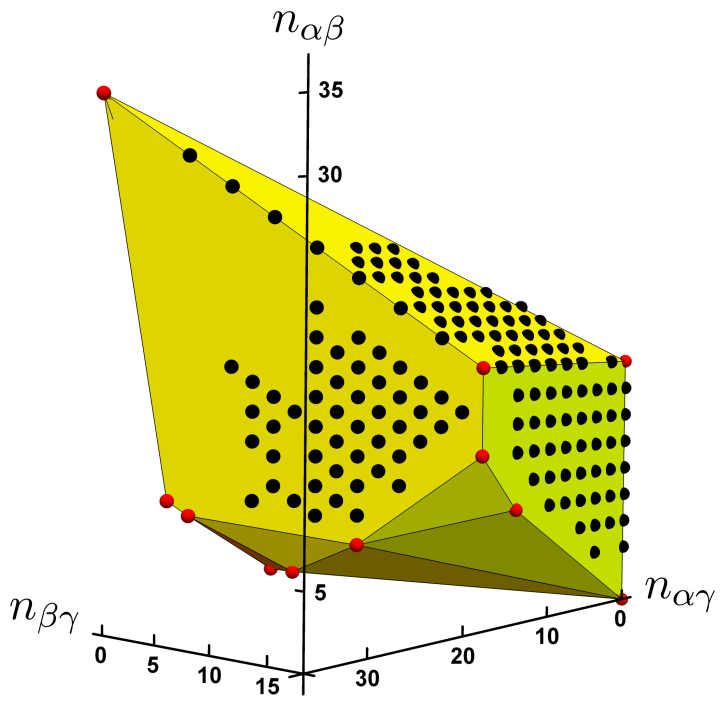


Figure 6.7: The convex hull of points in $\vec{n}$-space corresponding to the right tromino configurations generated by the 9×6 unit cell are shown. Here, we are considering a reduced interaction space $(n_{\alpha\beta}, n_{\alpha\gamma}, n_{\beta\gamma})$, as explained in Figure 6.6.

space. This is because generally multiple points of the full interaction space are mapped to the same point of the reduced space. To be specific, points of the full interaction space that are separated by a vector of the kernel of the transformation to the reduced space will be mapped onto the same point in the reduced space.

The point $(n_{ab}, n_{ac}, n_{bc}, n_{ad}, n_{bd}, n_{cd}) = (0, 0, 36, 36, 0, 0)$ is a vertex point of the full interaction space corresponding to a single configuration of wallpaper group cm, illustrated in Figure 6.8(a). This point remains a unique vertex point $(n_{\alpha\beta}, n_{\alpha\gamma}, n_{\beta\gamma}) = (36, 36, 0)$ when transformed to the reduced space. However, the majority of unique vertex configurations become degenerate under the transformation to the reduced system. For example, the vertex point $(0, 8, 6, 8, 6, 14)$ of the full interaction space corresponds to a single configuration of pg wallpaper group, shown in Figure 6.8(b). This point is mapped to $(14, 14, 14)$ in the reduced space, where it is strictly within the reduced convex hull and has degeneracy of 2062 configurations, due to 70 other points from the full interaction space being mapped to the same point. This is an example of a configuration that could be designed as ground state using the full set of interaction parameters, but not the reduced set.

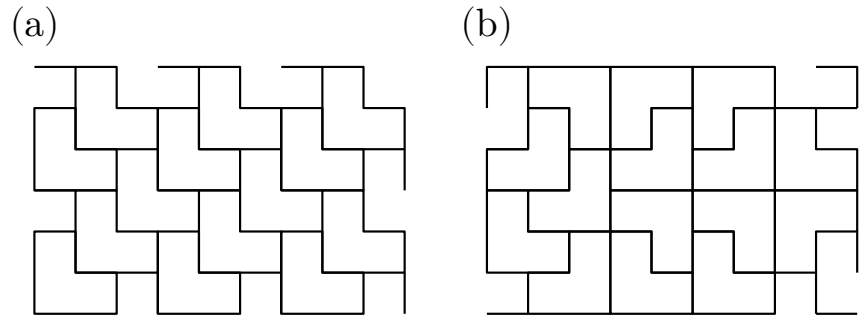


Figure 6.8: Right tromino configurations on the 9×6 unit cell. (a) $\vec{n} = (0, 0, 36, 36, 0, 0)$ is a unique vertex configuration of both the full and reduced interaction spaces. (b) $\vec{n} = (0, 8, 6, 8, 6, 14)$ is a unique vertex configuration of only the full interaction space.

Considering the boundary components more generally, an x-dimensional boundary component of the reduced convex hull corresponds to a y-dimensional boundary component of the full convex hull, with $y \geq x$. This is because lowest energy states that are possible for the reduced space are always possible for the full space too. As an example, the edge of the reduced convex hull defined by $n_{\alpha\gamma} = 0$, $n_{\beta\gamma} = 18$ contains 9 points corresponding to 2,772 right tromino configurations. In the full space, this line corresponds to a 3-dimensional highly degenerate boundary component of the full convex hull. The corresponding configurations are made up of many copies (possibly rotated or translated) of the second arrangement of right trominoes shown in Figure 6.5. The interactions between these motifs determine the lowest energy state. In the reduced space, there is only 1 remaining degree of freedom $\varepsilon_{\alpha\beta}$, while in the full space, there are 3: $\varepsilon_{ab}, \varepsilon_{ac}, \varepsilon_{bc}$.

6.4 T-Tetromino

T-tetrominoes have 6 face types; thus, the interaction count space is 15-dimensional (after taking into account the packing constraints - to eliminate the same face type interactions). However, there are additional packing constraints, similar to the right tromino packing problem. In this case, the effective interaction space is reduced to 10 dimensions, due to 5 packing constraints:

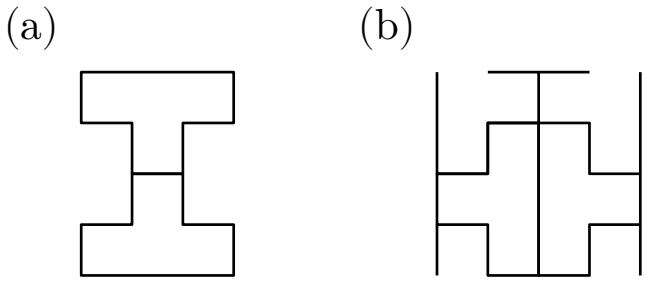


Figure 6.9: (a) The entropically disfavoured interaction f-f between T-tetrominoes. (b)The only T-tetromino configuration on the 4×4 unit cell with $n_{ff} \neq 0$.

$$\begin{aligned} n_{ad} &= 0, \\ n_{ae} &= 0, \\ n_{be} + n_{de} &= n_{cd}, \\ n_{df} + 2n_{cd} + n_{ce} &= n_{be} + NM/2, \\ n_{be} + n_{ef} &= n_{cd}, \end{aligned} \tag{6.2}$$

where $N \times M$ are the dimensions of the unit cell. In addition to these hard packing constraints, there is one soft constraint $n_{ff} = 0$. By soft constraint, we mean the majority of configurations adhere to the constraint, but not all. For the T-tetromino, the interaction type f-f occurs when the middle ends of two T-tetrominoes are in contact, as shown in Figure 6.9. The remaining space around these two T-tetrominoes is quite difficult to pack fully, so this motif is quite rarely seen in the high entropy state.

For example, the 12×10 unit cell has 87.7% of T-tetromino configurations with $n_{ff} = 0$. One of the unusual configurations with $n_{ff} \neq 0$ is shown in Figure 6.10. The high degeneracy of the n_{ff} plane can be seen in Figure 6.11, where we have chosen to compare n_{ff} with n_{ab} and n_{bc} for the visual comparison in 3 dimensions.

To quantitatively understand how closely the soft constraint is obeyed, we have calculated the average value of n_{ff}/NM over the set of configurations of the unit cell with dimensions $N \times M$, for the $12 \times (10, 8)$, 10×8, $8 \times (8, 6, 4)$, 6×4, and 4×4 unit cells. The notation here $N \times (A, B)$ means $N \times A$ and $N \times B$. We plot $\langle n_{ff} \rangle / NM$, which is the average value of n_{ff}/NM, against NM in Figure 6.12. As can be seen in the figure, the value of $\langle n_{ff} \rangle / NM$ becomes small for larger unit cells, where the ensemble of configurations begins to approximate the fully random limit of T-tetromino tilings. However, $\langle n_{ff} \rangle / NM$ is also quite small for all the

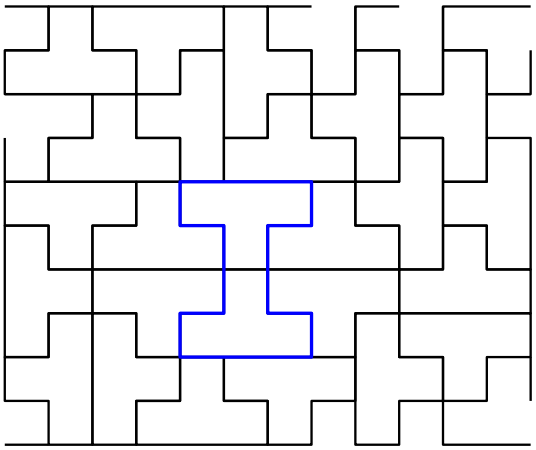


Figure 6.10: Example configuration of T-tetrominoes on the 12×10 unit cell. There is one f-f interaction, between the two T-tetrominoes highlighted in blue.

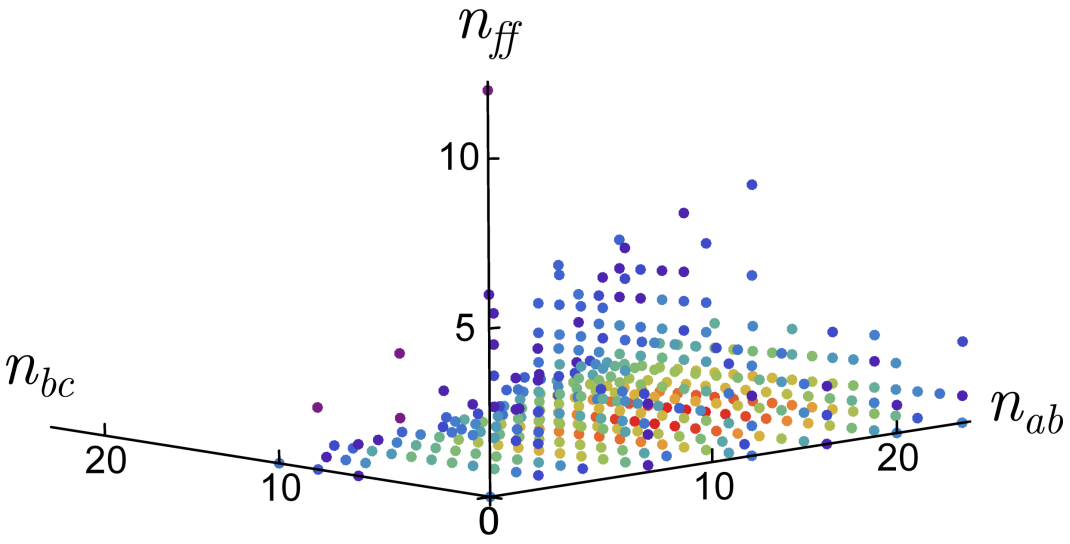


Figure 6.11: Degeneracy in coordinates (n_{ab}, n_{bc}, n_{ff}) for T-tetromino configurations of the 12×10 unit cell. The logarithm of degeneracy is used as a colour function for the points so that red means high degeneracy and blue means low. Most of the red points lie in the $n_{ff} = 0$ plane.

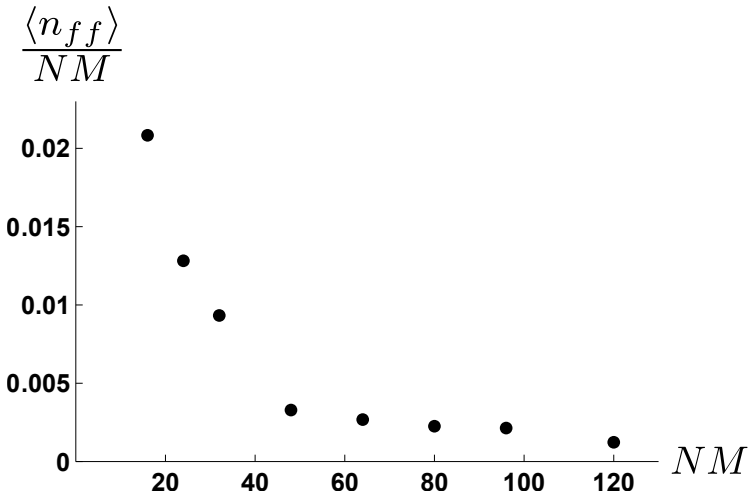


Figure 6.12: $\langle n_{ff} \rangle / NM$ is plotted against NM, for each of the unit cells $12 \times (10, 8)$, 10×8, $8 \times (8, 6, 4)$, 6×4, and 4×4.

unit cells tested. The low values of $\langle n_{ff} \rangle / NM$ justify the term soft constraint to describe the subspace $n_{ff} = 0$. The relatively infrequent f-f interactions for random packings can also be thought of as due to an entropic force, which disfavours the f-f interaction. We expect there are similar pseudo-constraints for more complicated polyominoes also.

In conclusion for this chapter, we have shown how the convex hull construction and other methods used for the domino system can be applied to more general polyomino shapes. Furthermore, we see some phenomena in common between the domino system and other polyominoes. High degeneracy faces of the convex hull allow for the possibility of high degeneracy ground states, which may be traversed by small rearrangements. Straight trominoes parallel yet shifted with respect to each other result in less well connected configurations. Also, an edge of the convex hull of right tromino configurations corresponds to a subset of configurations with a particular motif. Other phenomena are new for polyominoes, which was not seen for the domino system. Straight trominoes have an easy axis in $\vec{n}$-space due to shifted positions. T-tetrominoes and right trominoes have additional packing constraints. Many interaction parameters mean that defining reduced interactions is an invaluable tool. Finally, there are strong entropic forces for T-tetromino configurations.

Chapter 7

Random DLX

For larger unit cell sizes, the direct enumeration of configurations becomes infeasible. In this chapter, we describe a modification to algorithm DLX that is used to randomly sample polyomino configurations. This allows for the exploration of larger unit cells that cannot be enumerated in reasonable time with algorithm DLX. Like algorithm DLX, our random DLX algorithm has the benefit that it can be used with polyominoes of general shape and explores the full set of possible configurations on an $N \times M$ rectangular region with periodic boundary. This is in contrast to the sampling algorithms of Ref. [45] and Ref. [46] mentioned in Chapter 1, which can only sample domino configurations and do not sample the full set of domino configurations on a rectangular region with periodic boundary.

We define a Markov chain on the tree of partially completed configurations in order to sample fully-packed configurations with equal probability. The statistics of the 8×8 unit cell are used to test the algorithm, since this unit cell is small enough for the enumerative algorithm DLX. We then explore some of the properties of the larger unit cells and compare the peak of degeneracy with a theoretical value we calculate for the limit of large unit cells. More generally, random DLX can be used to sample the solutions of any exact cover problem with uniform probability. However, in this thesis, we consider only the polyomino problem.

7.1 Random DLX Method

Our aim is to extend algorithm DLX to be able to sample configurations randomly. However, the naïve extension fails to sample with uniform probability. In the naïve extension, we add polyominoes to the partial solution randomly. Specifically, after selecting a square to be covered, we choose a polyomino to cover that square with

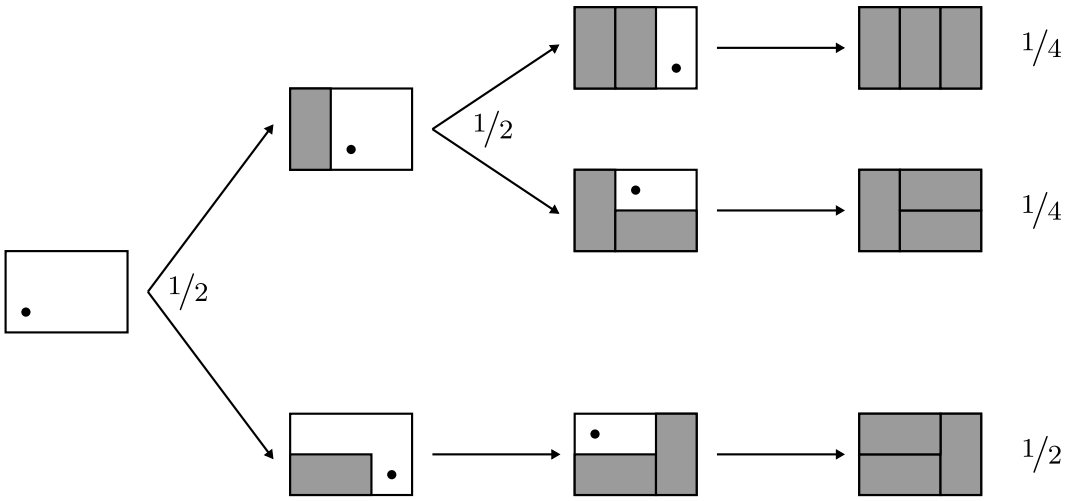


Figure 7.1: A straightforward extension of algorithm DLX for random sampling, where naïvely assigning equal branch probabilities at each step leads to unequal probabilities of sampling the three final configurations. The 3×2 region with rigid boundary is packed with dominoes, shown with grey fill. The dot shows the next square to be covered. The choice used here was to cover lowermost, then leftmost squares. Where there are two ways to cover the square, the algorithm is equally likely to continue on either branch. Note that in the specific case shown here, there happens to be no backtracking of algorithm DLX.

uniform probability from the possible polyominoes that could cover it. However, as can be seen in the example of Figure 7.1, the completed configurations will appear with biased probabilities. So, this method could be used to select a random completed configuration, but not with uniform probability from the set of all completed configurations. Therefore, we must use a different method in order to sample from configurations with uniform probability. The rest of this section is centred on the description of the algorithm we have found to meet this requirement.

When picturing the structure of algorithm DLX or our random DLX, it is helpful to think of each partially completed polyomino configuration as a node in a tree, which is linked to child nodes in front of it and a parent node behind it. The child nodes are made by adding a polyomino, and the parent node by taking away a polyomino. This is shown in Figure 7.2. Nodes at the same depth in the tree are said to be at the same level L. The node at the zeroth level is the empty partial configuration (call it the root node R), and the nodes at the final level are the completed polyomino configurations. We denote the final level as F, which has value equal to the number of polyominoes in a fully-packed configuration. The nodes with no children are called leaf nodes, of which the final level nodes are a subset.

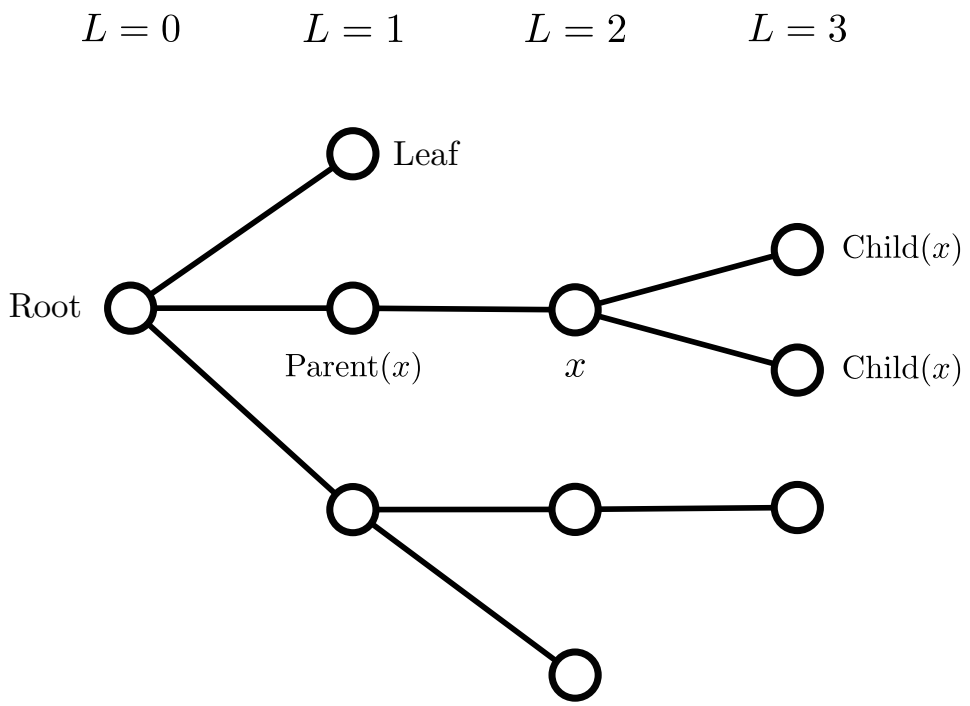


Figure 7.2: An example tree data structure for an exact cover problem. For our purposes, each node can be taken to represent a partially completed polyomino configuration, and links exist between partial configurations that differ by 1 polyomino. The forwards direction of the tree is to the right. The root node and an example leaf node are indicated. One example node x is labelled, along with its parent and child nodes. The partially completed configurations containing L polyominoes are at depth (or level) L. In this case, the final level F has value 3. All nodes at the final level are leaves corresponding to completed polyomino configurations.

The unmodified algorithm DLX is a depth-first recursive enumerative algorithm that begins at the root of the tree and continues forward along the tree structure by adding polyominoes, only backtracking when all nodes further forward along the tree have been reached. In contrast, for random DLX, there is a nonzero probability to backtrack to the parent node or continue forward to a randomly chosen child node. Our transition probabilities only depend on the currently occupied node, giving a discrete time finite Markov chain. When the Markov chain reaches a completed configuration at level F, the sample statistics are updated. By choosing the transition probabilities in the right way, we are able to sample the final level leaves with equal probability, giving correct sample statistics for the completed polyomino configurations. The probability of being at a partially completed configuration does not need to be uniform, since they are not used for sampling.

Here we define transition probabilities between partially completed polyomino configurations so that the completed configurations are sampled with uniform probability. To do this, we introduce parameters Q_L that control the rate of backward transitions from nodes at the Lth level. For a given node x at level L and having $C(x)$ child nodes, we use transition probability $Q_L/(Q_L + C(x))$ to move backwards to the parent node and $1/(Q_L + C(x))$ for the probability to make a forward transition to a particular child node. Note that there is equal probability to transition to each of the child nodes and there is zero probability to remain at the same node. The transition probabilities from one example node are shown in Figure 7.3. Furthermore, note that all nodes at level L have the same parameter Q_L.

The root node and the leaf nodes are special cases in that they have no parent node and no child nodes, respectively. However, the previously stated transition probabilities still apply, as long as we define $Q_0 = 0$ so there is zero backwards transition probability from the root node. For a leaf node x, the value of $C(x)$ is zero, so the backwards transition probability is calculated to be 1, as it should be.

We find the probability distribution for this Markov chain by using the detailed balance conditions. Consider 3 nodes y, x, and z such that x is a child node of y and z is a child node of x. This is illustrated along with transition probabilities in Figure 7.4. We use π_i to denote the probability to be at node i and use L for the level of node x. Looking at these 3 nodes, detailed balance gives 2 equations

$$\begin{aligned} \frac{\pi_y}{Q_{L-1} + C(y)} &= \frac{\pi_x Q_L}{Q_L + C(x)}, \\ \frac{\pi_x}{Q_L + C(x)} &= \frac{\pi_z Q_{L+1}}{Q_{L+1} + C(z)}, \end{aligned} \tag{7.1}$$

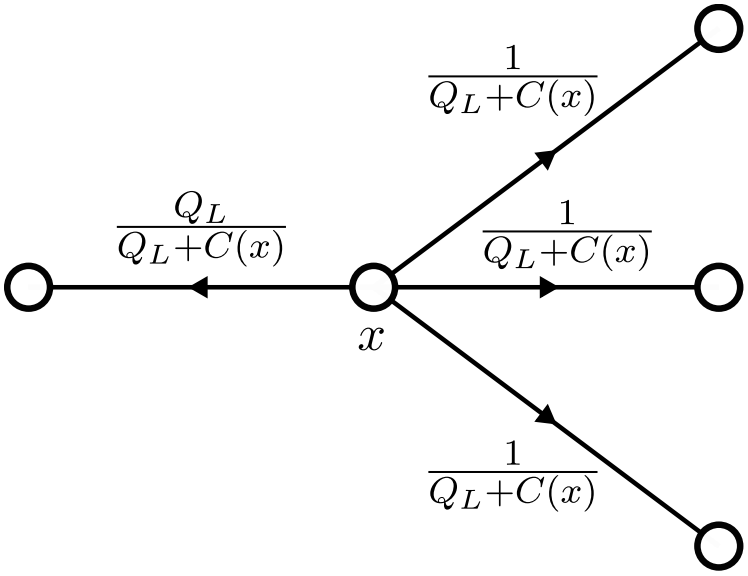


Figure 7.3: An example node x at level L in the tree of partially completed polyomino configurations. The transition probabilities from this node are shown. In this case, node x has 3 child nodes, so $C(x) = 3$. Q_L is an algorithm parameter that is equal for all nodes at level L.

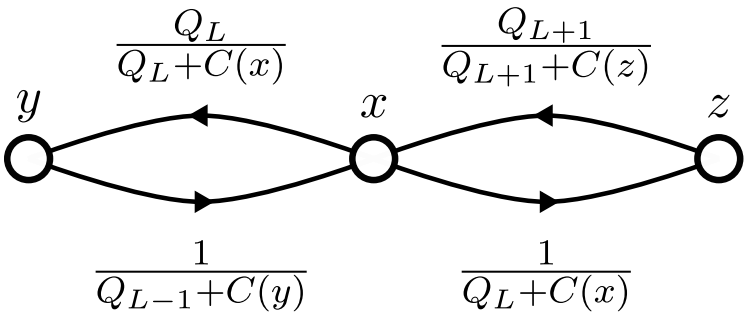


Figure 7.4: 3 Consecutively linked nodes from the tree of partially completed polyomino configurations are shown. Nodes y, x, and z are at levels $L-1$, L, and $L+1$, respectively. Other linked nodes are not shown. Alongside the links are the transition probabilities between nodes.

which are obtained by considering detailed balance between nodes y and x; and between nodes x and z. These two equations can be combined to eliminate π_x, resulting in

$$\frac{\pi_y}{Q_{L-1} + C(y)} = Q_L \frac{\pi_z Q_{L+1}}{Q_{L+1} + C(z)}. \tag{7.2}$$

This equation shows there is a factor Q_L introduced by skipping over the node at level L. A similar calculation can be done with the node z and one of its child nodes, introducing a factor Q_{L+1}. Repeating this calculation inductively, a factor is introduced for each level between the nodes. Furthermore, the root level has $Q_0 = 0$, and any node in the final level has no child nodes. Therefore, we end up with an equation relating the probability of the root node R with the probability of an arbitrary node k at the final level F :

$$\frac{\pi_R}{C(R)} = \left(\prod_{a=1}^{F-1} Q_a \right) \pi_k. \tag{7.3}$$

The final level nodes are the completed polyomino configurations; therefore, we obtain a simple relation for the probability of the Markov chain to arrive at any one of the completed configurations. Moreover, each of the completed configurations have equal probability.

We have defined transition probabilities and used the detailed balance condition to calculate probabilities for the Markov chain to arrive at the completed polyomino configurations. However, we must now show that the detailed balance condition provides the limiting probability distribution for the completed configurations.

For a finite Markov chain defined on a tree, there will always exist a distribution that satisfies detailed balance [100]; therefore, our procedure always results in a valid probability distribution. A distribution satisfying detailed balance is always a stationary distribution. Since the original algorithm DLX is made up of only one search tree, the Markov chain is irreducible and finite over the nodes of partially completed configurations. Therefore, our probability distribution is the unique stationary distribution. Our Markov chain has period 2 because it takes an integer multiple of 2 steps in the Markov chain to return to the same node. However, by considering just the nodes representing completed configurations, we obtain an aperiodic Markov chain because there is a nonzero probability to return to the same completed configuration before reaching another completed configuration. Since the Markov chain over the completed configurations is aperiodic, our stationary distri-

bution is the unique limiting distribution over the completed configurations.

The unique limiting distribution we have found has equal probability for each completed polyomino configuration. Therefore, by taking a sample from each completed configuration that the Markov chain reaches, it is possible to sample the completed polyomino configurations with uniform probability. It is important that Q_L must not be set to zero for any level except the zeroth level; otherwise, not all completed configurations would be connected.

In the Markov chain we have defined, each ith level parameter Q_i can be freely chosen as any positive number, apart from the zeroth level parameter, which must equal zero. The parameters Q_i control how likely the Markov chain will move back along the tree from level i. Therefore, these parameters determine how the Markov chain explores the space of polyomino configurations. For the ideal Markov chain, we would like to have the smallest possible correlation between successive samples while also being able to generate samples quickly. One could adjust each of the Q_i parameters individually to make an optimal Markov chain for a given exact cover problem. However, we make the simplification that all parameters Q_i are equal. The more general case is an interesting problem for future work.

Since we are choosing Q_i to be equal for all levels i, we can use the notation Q from now on. Equation (7.3) simplifies to

$$\frac{\pi_R}{C(R)} = Q^{F-1}\pi_k. \tag{7.4}$$

The probability of being at a completed configuration relative to the probability of being at the root node now only depends on the single parameter Q. For larger values of Q, the Markov chain is more likely to move back along the tree towards the root node, meaning successive samples of completed polyomino configurations are less correlated; however, if Q is too large, the time between successive samples may be too great to be practical.

It is important to find a value for Q that balances the needs of uncorrelated samples with the time to generate samples. A reasonable guess for a good value of Q can be found by making the probability of the final level equal to the probability of the zeroth level. From there, it is possible to adjust Q further. The probability of the zeroth level is $P_F = \pi_R$ because the only node at the zeroth level is the root node. The number of nodes in the final level is equal to the number of polyomino configurations that can be generated on the $N \times M$ unit cell, which we denote N_F. Since each of the nodes at the final level has probability equal to π_k, the probability of being at the final level is $P_F = \pi_k N_F$. Using Equation (7.4), we obtain the

equation

$$\frac{P_R}{C(R)} = \frac{Q^{F-1}}{N_F} P_F, \tag{7.5}$$

relating the probabilities for the Markov chain to visit the zeroth and final levels. For the probability of the zeroth and final levels to have comparable values, we require Q^{F-1} and N_F to be of comparable value. This means we have a reasonable probability to transition between complete configurations whose only common ancestor is the root node. The number of domino configurations that can be generated on a unit cell of area $2F$ has a known value in the limit of large F, when both unit cell sides tend to infinity for rigid or periodic boundary conditions [35]

$$\exp(2FG/\pi), \tag{7.6}$$

where G is Catalan's constant, approximately equal to 0.916. When using random DLX on large finite unit cells, N_F is approximately equal to Equation (7.6). Therefore, substituting this value in to Equation (7.5), we obtain

$$P_R = \frac{C(R)}{Q} \left(\frac{Q}{\exp(2G/\pi)} \right)^F P_F, \tag{7.7}$$

giving $\exp(2G/\pi)$ as a value for Q that can be used so that there is comparable probability to be in the final level or at the root node. The value $\exp(2G/\pi)$ is the number of arrangements per dimer [35]. In other words, the average number of child nodes in our tree of partially completed domino configurations. By setting Q equal to this value, we ensure that each node's probability is on average equal to the sum of probabilities of its child nodes; thus, keeping a balanced probability across different levels of the tree. A reasonable value of Q can be found for other polyominoes apart from the domino by using the number of arrangements per polyomino. Also, a good value of Q can be found for any exact cover problem by using knowledge of the tree structure. In the rest of this chapter, we test out our algorithm on fully-packed domino configurations for several unit cell sizes. Although Equation (7.7) contains an exponent of the number of dominoes F, the choice $Q = \exp(2G/\pi)$ allows for a consistent algorithm over different unit cell sizes. This is especially important for larger values of F, where a value of Q that is far from $\exp(2G/\pi)$ will result in a radically different probability being assigned to the different levels of the tree.

7.2 Random DLX Tests

Since we have enumerated the domino configurations on the 8×8 unit cell, we have an opportunity to test our random DLX algorithm for this unit cell size. Since random DLX takes a sample of configurations, we obtain only the relative frequency of configurations. This is in contrast to the usual algorithm DLX, which provided the absolute number of configurations because it enumerates all the configurations. However, many kinds of statistics can be gathered on a set of domino configurations without knowing their absolute number.

To test our algorithm, we sample one particular quantity and compare with the exact value that is known for the 8×8 unit cell. The quantity we have chosen is the number of configurations with p1 wallpaper group, as a fraction of the total number of configurations sampled. We denote this quantity F_{p1}. The exact value is provided by the usual (enumerative) version of algorithm DLX, which generates a total 1,224,518 configurations for the 8×8 unit cell, out of which there are 1,181,552 with p1 wallpaper group. Therefore the true value of F_{p1} is 0.9649 to four significant figures.

Our first test of random DLX is using $Q = 1.79163$, which is equal to $\exp(2G/\pi)$, rounded up by the fifth significant figure. We make this choice because in the worst case, we would prefer to sample few configurations, rather than sample many highly correlated configurations, especially since analysing domino configurations can take considerable computational resource. Using this value of Q, we sample configurations of the 8×8 unit cell, updating the sample statistic for F_{p1}. We do this over 10 runs, in each case printing the value of F_{p1} at time intervals of 0.1 seconds, which typically corresponds to 150,000 steps in the Markov chain. The values from the 10 runs are used to form a mean value at each time interval. The standard error of the sample mean over the estimates from the 10 runs gives an error value for the mean. The result is shown in Figure 7.5. The true value of F_{p1} is also shown for comparison. As can be seen in the figure, the random DLX algorithm quickly begins to converge to the correct value. In contrast, the usual algorithm DLX obtains a full enumeration of the configurations of the 8×8 unit cell on the order of hours.

For our second test, we use random DLX with several values of Q to investigate our previous hypothesis that $Q = \exp(2G/\pi)$ is a good value to choose. For this test, we use a slightly different quantity E_{p1}, which can be thought of as the absolute value of the fractional error of F_{p1}, compared to the true value. This new quantity tends to zero as the sample quantity F_{p1} approaches the true value, making

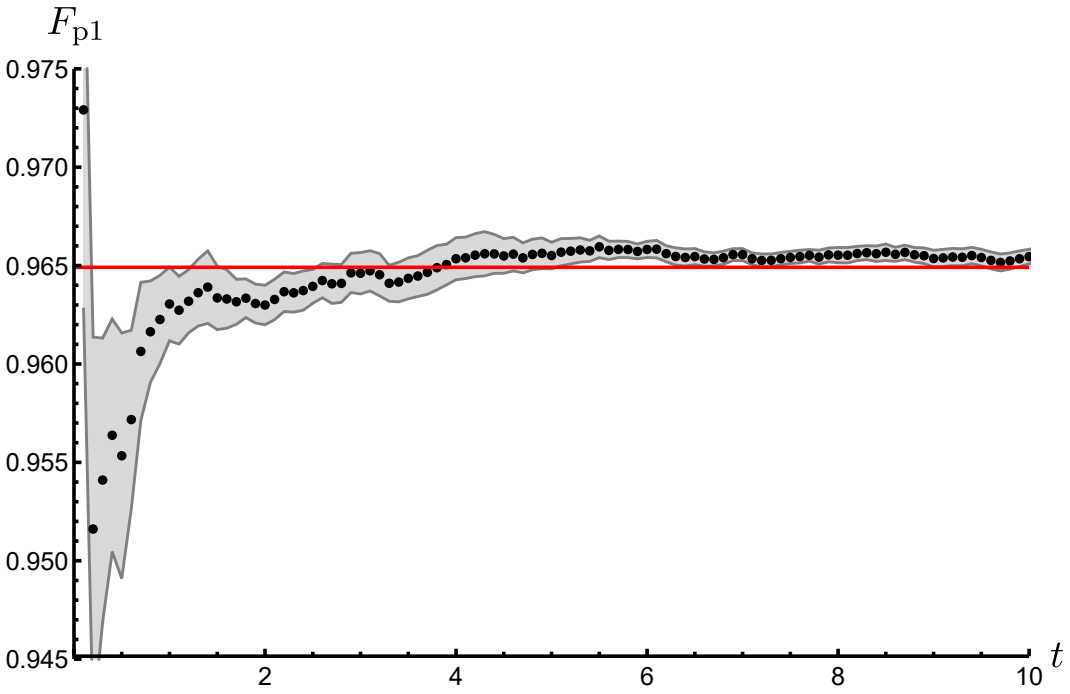


Figure 7.5: The fraction of p1 configurations F_{p1} is plotted against time (in seconds). The values (dark circles) and error (grey region) are obtained by taking 10 runs of the random DLX algorithm with $Q = 1.79163$. The horizontal red line indicates the true value of the fraction of p1 configurations : 1181552/1224518.

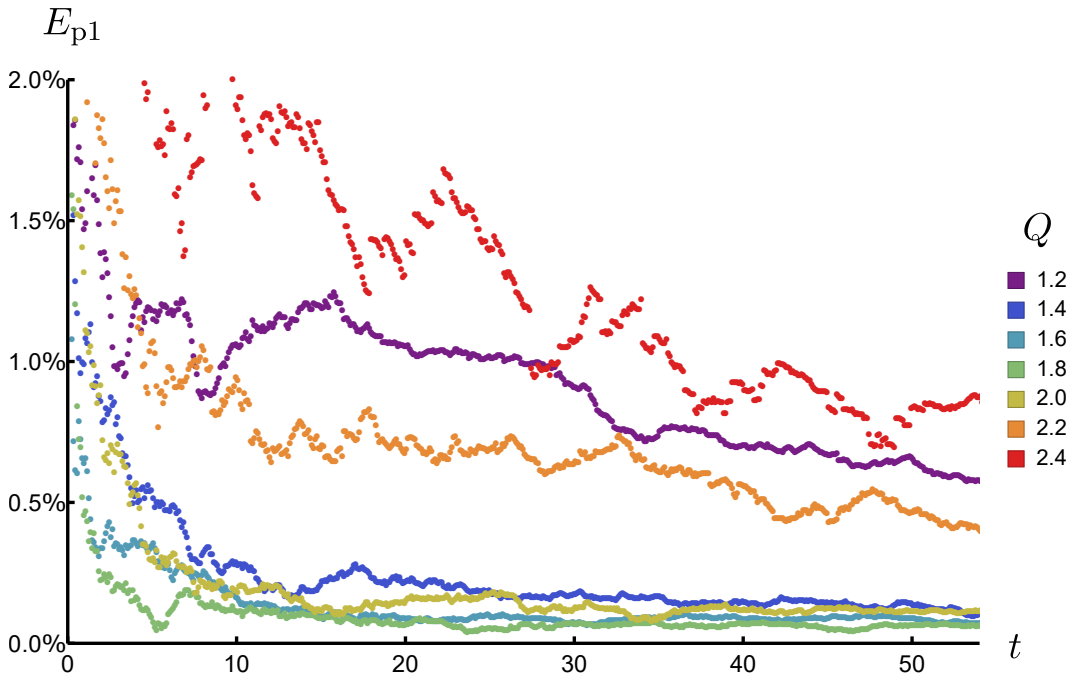


Figure 7.6: The absolute value of fractional error of the fraction of p1 configurations E_{p1} is plotted against time (in seconds). The points of a particular colour correspond to the mean over 10 runs of the random DLX algorithm using a single Q value.

it easier to distinguish the effectiveness of algorithms with different Q values. The new quantity E_{p1} is calculated as

$$E_{\mathrm{p1}} = \left| F_{\mathrm{p1}} \frac{1224518}{1181552} - 1 \right|. \tag{7.8}$$

The fractional error E_{p1} is plotted against time in Figure 7.6 for several Q values. Again, we print E_{p1} at regular intervals of 0.1 seconds. For each Q value, 10 runs of the algorithm are made, to give a mean value for E_{p1}. This gives a better indication of a typical run of the algorithm and allows for a smoother graph, which can be more easily interpreted.

In Figure 7.6, each of the fractional errors E_{p1} are decreasing with time. However, the fractional errors for algorithms with Q closer to $\exp(2G/\pi)$ decrease at a faster rate than the others. In particular, the extreme values $Q = 1.2$ and $Q = 2.4$ perform the worst. The best performing Q values are 1.6, 1.8, and 2.0. This supports our previous argument that low Q values generate highly correlated configurations, and high Q values generate too few configurations. In either case, the sample statistics collected by random DLX are worse when using an extreme value of Q. Therefore, from here onwards, we use $Q = 1.79163$ for the random

DLX algorithm, as it is approximately equal to $\exp(2G/\pi)$ and shows good results compared to other Q values.

7.3 Larger Unit Cell Exploration

The random DLX algorithm can be used to find sample statistics for unit cells that are too large to enumerate by the usual algorithm DLX. One interesting quantity is the degeneracy of points in $\vec{n}$-space. Here, we estimate the degeneracy for several square unit cells ($M = N$) using random DLX. To be able to compare between unit cells, we make use of interaction count vectors divided by the area of the unit cell $\vec{n}/N^2$, giving an interaction count vector per area. The limit convex hull shape we identified in 5.5.2 provides the range of possible values of $\vec{n}/N^2$. The values n_{ab} and n_{ac} must be within the range $[0, 2/3]$, and n_{bc} must be within $[0, 1]$.

For sampling the degeneracy, we have used 10 runs for each unit cell size and allowed each Markov chain 30 minutes to collect sample statistics. We use $Q = 1.79163$ for random DLX on each unit cell, which we have previously explained is a good parameter value over different area unit cells. For each unit cell size, the degeneracy estimate can be made by combining the samples from all 10 runs, with an error given by the standard error of the sample mean.

In Figure 7.7, we show the degeneracy of points in the space of interaction count per area for square unit cells $N = M =$ 8, 10, 12, 14, and 16. The colour of points give the degeneracy; however, for the larger unit cells, the high degeneracy points are completely surrounded by the low degeneracy ones. The degeneracy of the 8×8 unit cell given by random DLX is very similar to the exactly enumerated degeneracy shown previously in Figure 5.22. Despite this, not all points were found, for example, the point $\vec{n} = (0, 0, 0)$. This point is a vertex of the convex hull of the exactly enumerated set of configurations, corresponding to a possible ground state. This illustrates a drawback of the random sampling algorithm: important configurations can be missed.

For the larger unit cells, the shape of the set of points becomes more amorphous, as more of the extremal points in $\vec{n}$-space are missed out. As an example, the shape of the convex hull of points for the 12×12 unit cell is known to be of exactly the shape shown in Figure 5.12, as explained in Subsection 5.5.2. However, the shape in Figure 7.7 is completely different. As the unit cell gets larger, the extremal points become more atypical and so less likely to be found by random DLX. Another point about using random DLX on larger unit cells is that very few of the sampled configurations have any symmetry because they are incredibly atyp-

ical. In the combined computational time of 10 hours of random sampling of the 14×14 and 16×16 unit cells, only configurations with p1 wallpaper groups were found. For these reasons, an analysis of domino configurations similar to that of the enumerative algorithm DLX is not made here. It may be possible to introduce a further acceptance/rejection criterion in the Markov chain to favour configurations with particular properties, such as low energy. However, a decision would need to be made for the partially completed configurations, which do not automatically inherit the same properties as the completed configurations. This is an interesting possibility that we have not explored.

As the unit cell size increases, the degeneracy of the extremal points in $\vec{n}$-space becomes small relative to the total degeneracy. Conversely, the degeneracy of the most typical points in $\vec{n}$/-space becomes large with unit cell size, relative to the total degeneracy. This can be inferred by the decreasing spread of sampled points in $\vec{n}/N^2$-space, seen in Figure 7.7. We show this more explicitly in Figure 7.8, where we plot the degeneracy projected onto the directions n_{ab}/N^2 and n_{bc}/N^2. The plot for n_{ac}/N^2 is almost identical to n_{ab}/N^2, due to symmetry, so it is not shown. The absolute value of degeneracy is not known, so we choose the degeneracy of all points to sum to 1. In this way, the plot can be thought of as an empirical probability mass function.

Figure 7.8 also includes Gaussian curves that have been fitted to the degeneracy. The degeneracy tends to a Gaussian shape with increasing unit cell area. This is because different locations within the unit cell are sufficiently uncorrelated so that the $\vec{n}$-vector of the overall configuration receives contributions from many uncorrelated regions. This gives a normally-distributed degeneracy by the central limit theorem. Figure 7.9 shows how the standard deviation of the approximate Gaussian varies with N. As shown in Figure 7.9, for large N, the inverse of the standard deviation is proportional to N, for projections of the degeneracy onto both n_{ab}/N^2 and n_{bc}/N^2. In the standard formulation of the central limit theorem, the inverse of the standard deviation would be proportional to the square root of the number of independent samples. Therefore, the number of effectively independent regions within the unit cell increases proportionally with N^2. This gives the intuitive result that the number of effectively independent regions is proportional to the total area. Furthermore, the constant of proportionality is greater for the n_{ab}/N^2 direction because there is a smaller range of values, compared to the n_{bc}/N^2 direction.

As well as the spread of the degeneracy in $\vec{n}/N^2$-space, the mean value of $\vec{n}/N^2$ weighted by degeneracy is also of interest. The peak of the approxi-

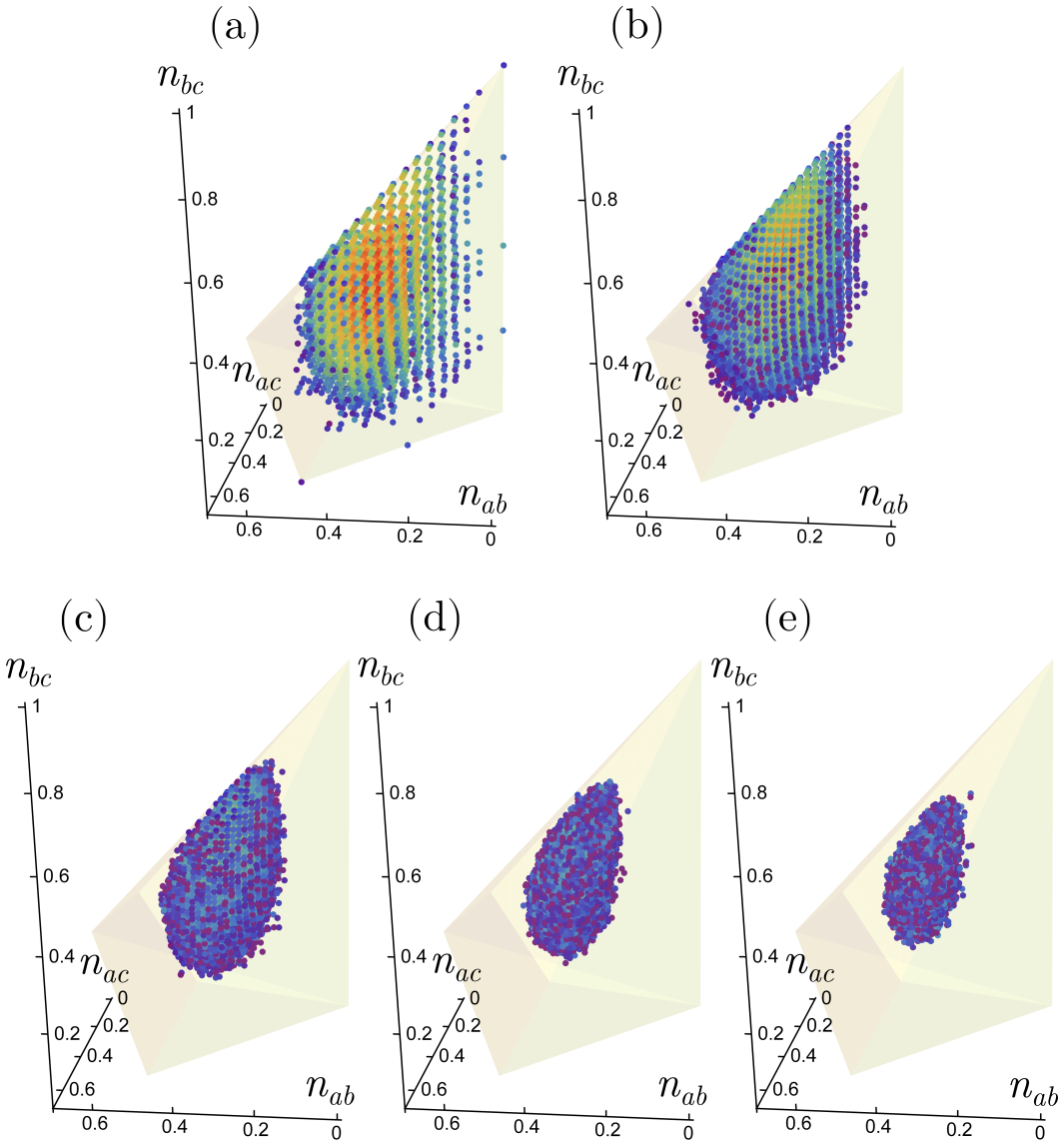


Figure 7.7: Points of $\vec{n}/N^2$-space corresponding to at least one domino configuration generated by random DLX on an $N \times N$ unit cell are shown as coloured dots, for N equal to 8, 10, 12, 14, and 16, for (a), (b), (c), (d), and (e), respectively. The dots are coloured according to the degeneracy, by a logarithmic colour scale. Red corresponds to the greatest value, and blue the least. The light yellow shadow indicates the limit convex hull for the set of all domino configurations, described in Subsection 5.5.2.

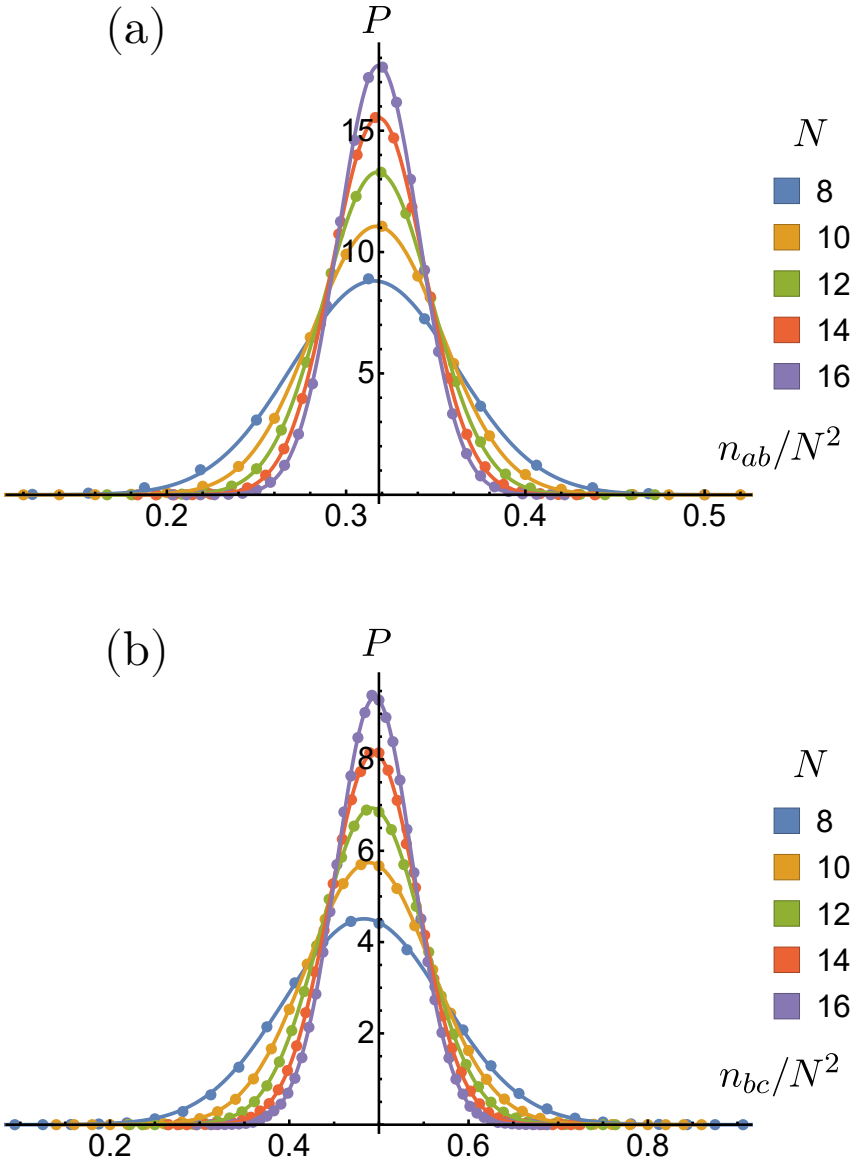


Figure 7.8: In (a) and (b), the empirical probability mass functions sampled by random DLX are shown by points plotted against n_{ab}/N^2 and n_{bc}/N^2. This is done for each of the unit cells $N = M = 8$, 10, 12, 14, and 16. Gaussian functions are fitted to each empirical probability mass function and plotted in the same colour.

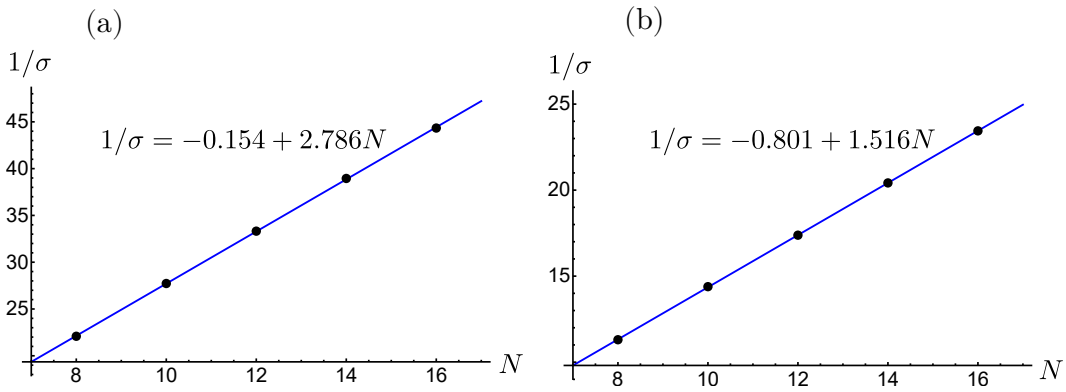


Figure 7.9: The standard deviation is shown for each of the fitted Gaussian curves given in Figure 7.8. This includes unit cells with $N = M = 8$, 10, 12, 14, and 16. (a) and (b) correspond to the standard deviation in the components n_{ab}/N^2 and n_{bc}/N^2, respectively.

mate Gaussians in Figure 7.8 give an indication of this. The mean value of $\vec{n}/N^2$ weighted by degeneracy is the point in $\vec{n}/N^2$ space that almost all configurations correspond to, for sufficiently large unit cells. The largest unit cell we have simulated here is the 16×16 unit cell. The mean value of $\vec{n}/N^2$ for this unit cell is $\vec{n}/N^2 = (0.31822, 0.31819, 0.4955)$, with error $(0.00008, 0.0005, 0.0002)$, which is again calculated by the standard error of the sample mean over the 10 runs made for this unit cell.

7.4 Limit Of Interaction Counts

Due to the importance of the limiting $\vec{n}/N^2$ vector of random configurations in the large unit cell limit, we would like to compare our estimated value with a theoretically calculated value. In fact, such a value can be calculated in the limit of large system size, using the values of domino correlations given by Ref. [47]. In that paper, the joint occupation probability for a pair of dominoes in a random domino configuration was calculated using correlations obtained by perturbing the Pfaffian that encodes the configurational grand partition function for non-interacting domino configurations. These joint occupation probabilities can be thought of as exact values for the entropic force between dominoes. In the limit of large system size, the joint occupation probabilities given by Ref. [47] are especially simple, since they only depend on the relative orientation and position of the two dominoes.

To calculate the theoretical limiting value of $\vec{n}/N^2$, we only need to make use of the occupation probabilities for dominoes that are in contact, since other domino

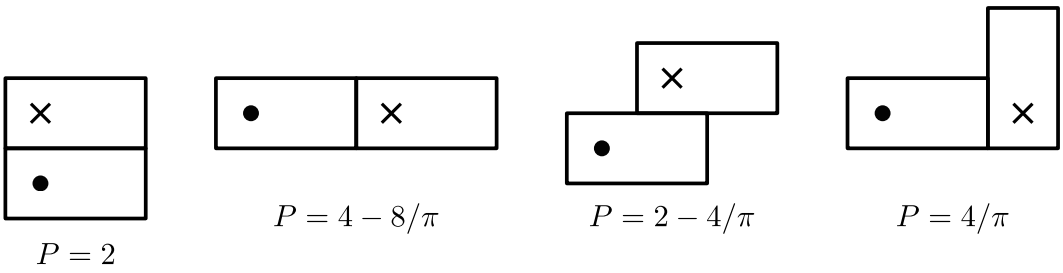


Figure 7.10: Relative probabilities are shown for pairs of dominoes in contact, for various relative positions and orientations. These values are calculated in Ref. [47] by considering the probability of two squares (indicated by the circle and cross) to be covered by dominoes of a particular orientation.

pairs do not contribute to the interactions $\vec{n}/N^2$. These occupation probabilities are given by Ref. [47], and are shown in Figure 7.10. For simplicity, we use relative probabilities, not absolute probabilities. It should be stressed that the probability shown in Figure 7.10 is the relative probability of the orientation of the dominoes, given they cover the two squares indicated as a circle and a cross. Note that the relative probabilities of all other orientations and relative positions of a domino pair in contact can be inferred by those shown in Figure 7.10, due to symmetry.

We now use the relative probabilities given by Ref. [47] to determine the limiting value of $\vec{n}/N^2$. It is important to properly count all the possible ways two dominoes can interact. To do this, imagine having a random domino configuration, then choosing two neighbouring squares. The orientations of the dominoes covering those two squares define the interaction type that occurs on the boundary between those two squares. Therefore, the probability of a randomly chosen interaction being of a particular type is equal to the probability of the two squares having dominoes oriented such that the particular interaction type occurs. All possible orientations of the pair of dominoes covering those points are shown in Figure 7.11, along with the associated interaction type that results from it. There is also the possibility of one domino covering both squares, but in that case there is no interaction, so we can ignore it.

The relative probability shown in Figure 7.10, along with the number of arrangements that lead to the same interaction type, shown in Figure 7.10, can be used to give the relative frequency of each interaction type. This is shown in Table 7.1. For example, the b-c interaction occurs $\pi/2$ times as frequently as the a-b interaction, in a random domino configuration on a unit cell in the limit of infinite cell size.

The relative frequency of interaction types gives the relative values of the

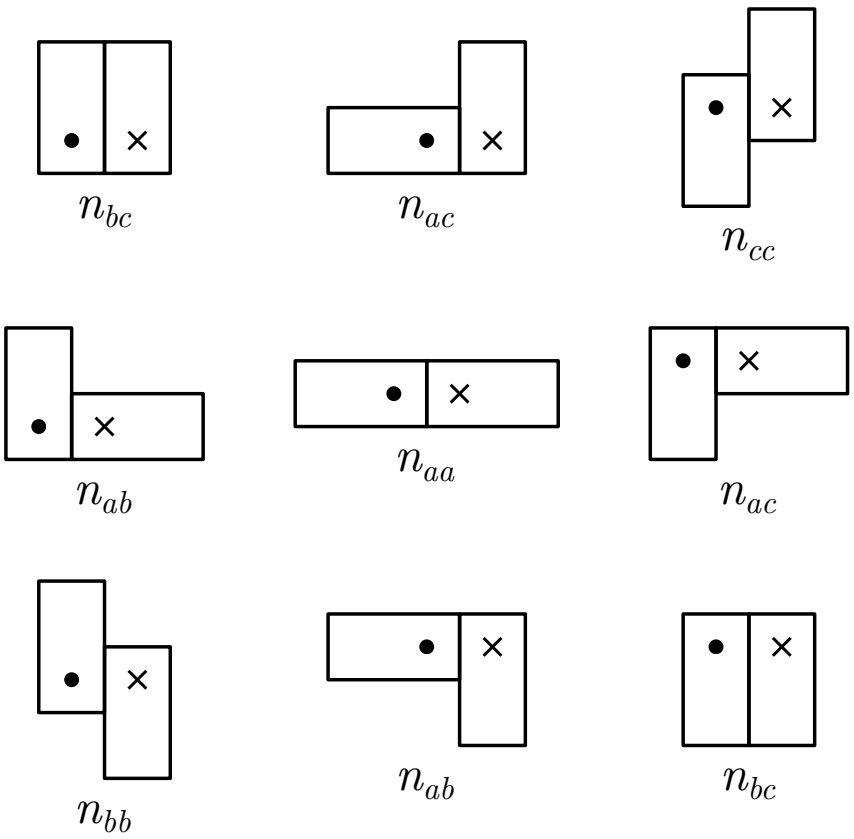


Figure 7.11: The possible arrangements of dominoes that cover two adjacent squares are shown, along with the type of interaction between those two squares. In each case, the two squares under consideration are indicated by a circle and cross.

interaction type	relative probability	arrangements	relative frequency
a-*a*	$4 - 8/\pi$	1	$4 - 8/\pi$
a-*b*	$4/\pi$	2	$8/\pi$
a-*c*	$4/\pi$	2	$8/\pi$
b-*b*	$2 - 4/\pi$	1	$2 - 4/\pi$
b-*c*	2	2	4
c-*c*	$2 - 4/\pi$	1	$2 - 4/\pi$

Table 7.1: The calculation of relative frequency of interaction types is shown as the product of relative probability and the number of possible arrangements.

interaction counts. For example, $n_{bc}/n_{ab} = \pi/2$. To obtain the absolute values, we can use a single packing constraint. The equation $NM = 2n_{aa} + n_{ab} + n_{ac}$ can be written as

$$NM = 2\frac{n_{aa}}{n_{ab}}n_{ab} + \frac{n_{ab}}{n_{ab}}n_{ab} + \frac{n_{ac}}{n_{ab}}n_{ab} = 2\frac{4 - 8/\pi}{8/\pi}n_{ab} + n_{ab} + n_{ab}, \tag{7.9}$$

by using the relative frequencies. This calculation gives $n_{ab} = NM/\pi$. The value n_{ac} is also NM/π due to symmetry, but can be checked to be equal. Finally, n_{bc} can be calculated as

$$n_{bc} = \frac{n_{bc}}{n_{ab}}n_{ab} = \frac{4}{8/\pi}NM/8, \tag{7.10}$$

so the value of n_{bc} is $NM/2$. Therefore, the interaction count per area of a random domino configuration on a unit cell in the infinite limit is given by $\vec{n}/(NM) = (1/\pi, 1/\pi, 1/2)$. Furthermore, the high entropy state for a sufficiently large unit cell consists of configurations with $\vec{n}/(NM)$ close to this value.

We can compare the exact theoretical value $\vec{n}/N^2 = (1/\pi, 1/\pi, 1/2)$ of the large square unit cell limit to our estimated values for finite unit cells. Figure 7.12 show the elliptical regions within 1 standard deviation from the mean for the Gaussian approximations of the degeneracy over several unit cell sizes. Only n_{ab}/N^2 and n_{bc}/N^2 are plotted, but n_{ac}/N^2 gives a very similar picture, due to symmetry. The straight red lines in the figure indicate the theoretical value for the mean of $\vec{n}/N^2 = (1/\pi, 1/\pi, 1/2)$. The Gaussian peak is still fairly wide even for the 16×16 unit cell, indicating that we are still quite far from the infinite cell size limit. The theoretical limit is well contained within the Gaussian peaks, which suggests that the limit may be being approached smoothly.

The mean value of $\vec{n}/N^2$ weighted by the degeneracy is a more specific way to show how far the random domino system is from the limit of infinite unit cell size. In Figure 7.13, this mean is shown for each of the unit cells used previously except the 8×8 cell, which is too far from the theoretical limit value. The theoretical limit value is shown by the straight red lines. The errors for each of the points are calculated by taking the standard error of the sample mean over 10 runs. As can be seen in the figure, the mean values of $\vec{n}/N^2$ are close to the theoretical limit and they get closer with increasing unit cell size. However, the error bars show that the values are definitely not yet equal to the theoretical limit value. In particular, a slightly higher fraction of n_{bc}/N^2 is entropically favoured in the infinite unit cell limit, compared to the unit cells we have sampled. In other words, the value of

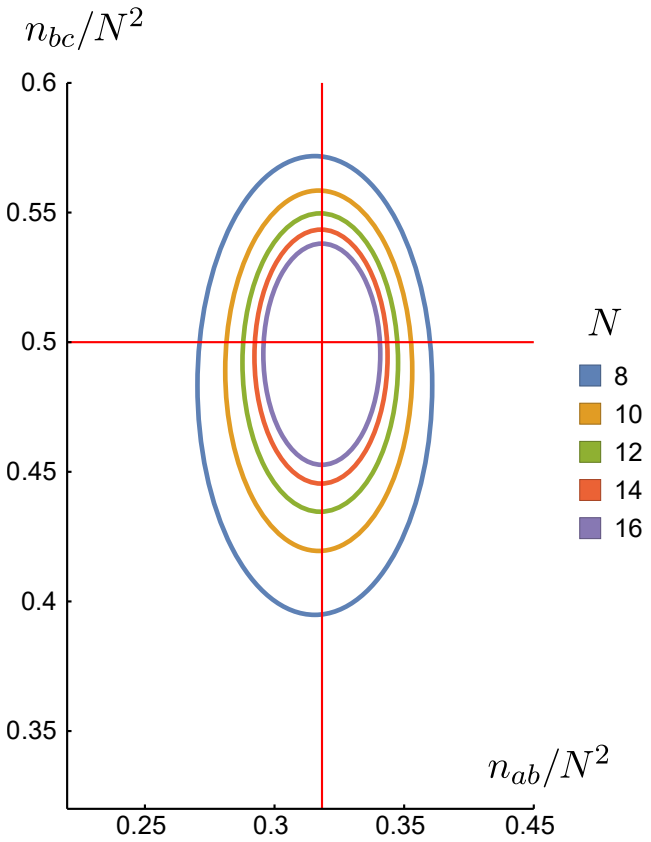


Figure 7.12: The regions of 1 standard deviation from the mean are shown for the Gaussian fit to the degeneracy of domino configurations generated by random DLX on unit cells of size $N = M = 8$, 10, 12, 14, and 16. The intersection of red straight lines shows the value $(1/\pi, 1/2)$ corresponding to the theoretical peak of degeneracy for large unit cell size.

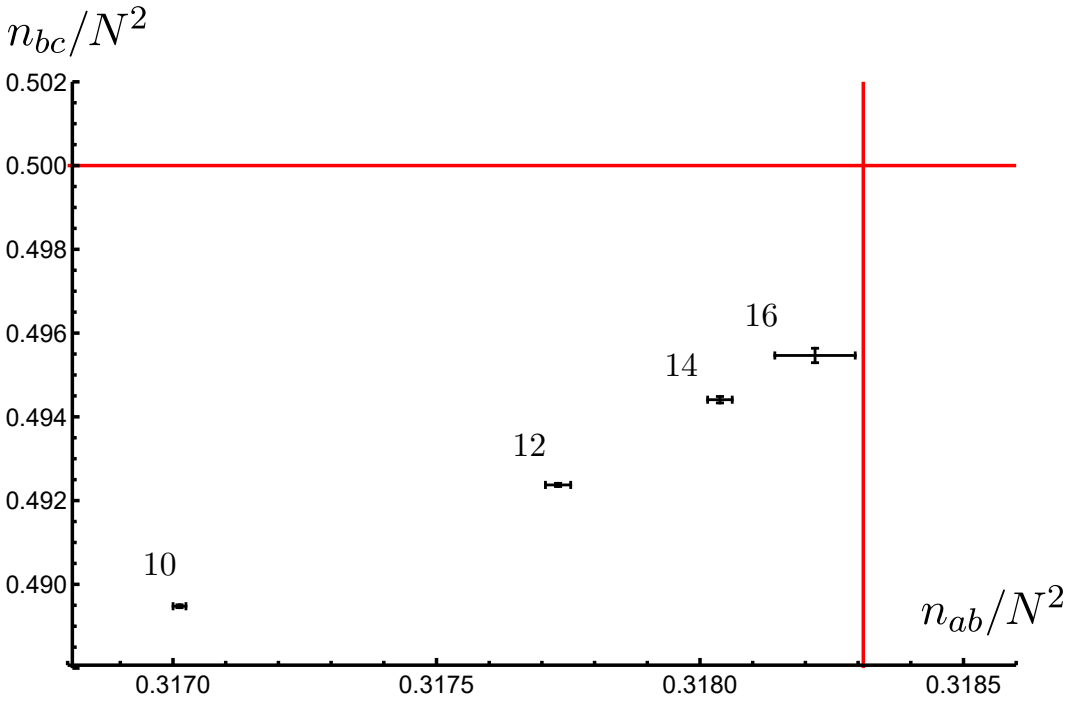


Figure 7.13: The mean value of $(n_{ab}, n_{bc})/N^2$ weighted by degeneracy is shown for domino configurations generated by random DLX on square unit cells of size $N = M = 10$, 12, 14, and 16. Error bars indicate the standard error of the sample mean over the 10 runs made for each unit cell. The intersection of straight red lines gives the theoretical limit of the mean of $(n_{ab}, n_{bc})/N^2$ for large unit cell size, having value $(1/\pi, 1/2)$.

n_{ab}/N^2 is approaching the limit value quicker than n_{bc}/N^2.

The random DLX algorithm we have described in this chapter has enabled us to gather statistics on larger system sizes using uniform probability over domino configurations. The algorithm we have developed as an extension to algorithm DLX also allows for uniform sampling over general exact cover problems; in the more general case, the algorithm would need to be tested to find an appropriate value for the backward transition parameter Q. We have also studied how the domino system tends to the limiting state of random domino patterns on the plane, identifying finite size effects in the approach to this limit.

Chapter 8

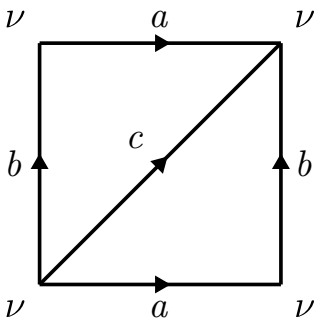
Fixed Defects

In an ideal experiment, the molecular network forms on top of a flat substrate, giving perfect molecular coverage. However, if there are some defects protruding upwards from the substrate, this causes gaps in the molecular network. In this chapter, we describe a molecular system that includes fixed substrate defects at periodic locations. We consider domino-shaped molecules, which are fully packed on the substrate, except at the locations of the defects. In this sense, the defects act as immobile vacancies in the domino tiling. Homology techniques can be used to understand the connectedness of domino tilings by local rearrangements in the presence of fixed defects, and we use these methods here to describe how periodic defects tend to decrease the connectivity of the domino tilings, which may be a desirable property for providing stability to a molecular network. Mathematical details and proofs for the homology constructions are given in Ref. [43].

8.1 Connectivity Of Configurations With Fixed Defects

Introducing fixed defects into our system changes the topology of the region, generally resulting in a different homology group. In previous chapters, we have described domino configurations on a unit cell with periodic boundary conditions, which has the topology of a torus with first homology group $\mathbb{Z}^2$. In that case, the homology class associated to any domino tiling can be determined by counting (with signs) the number of dominoes intersecting a horizontal and a vertical line, or cut, crossing the unit cell (and representing a basis for the homology). Here, we find different homology groups as a result of the defects and identify appropriate cuts for calculating the homology class.

When there are fixed defects in the unit cell, the translational symmetry



$$C_2 = 0 \qquad C_1 = xa + yb + zc$$

$$\partial_2(0) = 0 \qquad \partial_1(xa + yb + zc) = (x + y + z)(\nu - \nu)$$

$$\mathrm{Im}(\partial_2) = 0 \qquad \ker(\partial_1) = xa + yb + zc$$

$$H_1 = \ker(\partial_1)/\,\mathrm{Im}(\partial_2) = \mathbb{Z}^3$$

Figure 8.1: CW complex for the retract of the unit cell of even side length with 2 defects. Calculation of the homology group is given to the right. The C_i are the chain groups, ∂_i are the boundary operators and x, y, Z are integers.

group of the domino pattern must be a subgroup of the translational symmetries of the defects. In particular, domino configurations on a unit cell with a single defect only have the translational symmetry of the unit cell itself, which is equal to the spacing between defects on the periodic plane tiling. A unit cell made up of copies of the minimal unit cell can generate less symmetric domino patterns than the minimal unit cell, allowing for more complicated homology groups. Furthermore, we find that the connectivity of domino configurations depends on whether the defects are spaced by an odd or even number of squares.

8.1.1 Even Spaced Defects

The homology classes for a unit cell with one or more defects can be found in a similar way as for the unit cell with no defects. The difference of domino configurations defines the oriented transition graph between them and is a 1-cycle of the chain complex. This 1-cycle is the boundary of a 2-chain - integer combination of the plaquettes of the underlying square grid - exactly when the two domino configurations are related by a sequence of domino flip moves.

First, we describe the homology group for the unit cell with both side lengths even and containing two fixed defects. This space retracts to the CW-complex shown in Figure 8.1, from which its first homology group is easily deduced to be $\mathbb{Z}^3$.

To provide a method for determining the homology class of a particular domino configuration, we introduce a system of cuts as before and measure flows across them. Each cut must assign all boundary elements to the trivial homology class. The choice of cuts to use is related to a choice of representative cycles that provide a basis for the homology. The elements b, -a, and c-a-b can be used for

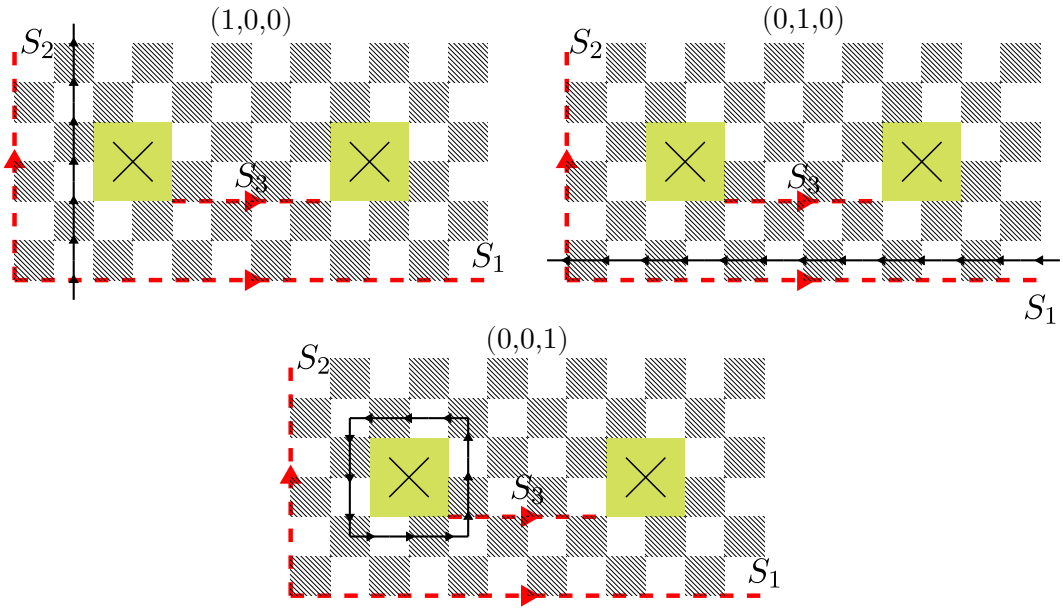


Figure 8.2: For the 12×6 unit cell with 2 defects of size 2×2, the three cuts are shown as red dashed arrows, giving values S_1, S_2, and S_3. Each unit cell shows one of the cycles that together generate representative cycles in each of the homology classes, as black lines with arrows. The sector (S_1, S_2, S_3) is displayed above each unit cell.

the representative cycles, see Figure 8.1, and cuts are chosen to assign these to the appropriate homology classes, as shown in Figure 8.2, along with the 3 representative cycles mentioned previously. The direction of each cut is shown explicitly, and the flow across the cuts is counted positively where the oriented transition graph passes through the cut from right to left. The homology class is given by a list of the values of the cuts (S_1, S_2, S_3), which we call the sector, analogous to the sector used to describe the unit cell without defects.

The first two cuts S_1 and S_2 are the same as for those of the unit cell without defects. As was explained in Subsection 2.3.4, a pair of domino configurations with different values for these cuts are not connected by local rearrangements, even when considering them as tiling the plane to become planar domino patterns. This remains true for planar patterns with fixed defects, so we denote the values of these cuts as global homology numbers. The third cut S_3 is a different kind of homology constraint. Figure 8.3 shows two example domino configurations that have different values of S_3. Despite the configurations being disconnected in terms of domino flip moves, there is a finite region around one defect that could be rearranged to transform one configuration into the other. This local rearrangement around the

defect changes the value of S_3; therefore, we call it a defect homology number.

The defect homology number is a weaker constraint than the global homology numbers because the former can be changed by a rearrangement of a finite number of dominoes. However, if the free energy of the molecular system strongly prefers molecules to be adsorbed to the substrate, the desorption-adsorption events would involve as few molecules as possible, which for domino shaped molecules is two. In this case, only domino flip moves are possible and the defect homology numbers are conserved. Otherwise, if several molecules are free to desorb at once, only the global homology numbers are important, and the connectedness of configurations is similar to that of the unit cell without defects. Interestingly, although the homology calculation does not depend on the size or shape of the defects, changing the defect homology number of larger defects would require desorption-adsorption events with a greater number of molecules, meaning larger defects give a stronger constraint.

In the general situation where there are $n \geq 1$ defects in a unit cell with even side length, the homology group is Z^{n+1}. To determine the homology class of a particular domino configuration we use $n-1$ cuts between the defects and 2 around the unit cell. Considering the connectivity network of domino patterns linked by domino flips, sets of domino patterns with larger periodicity are broken up into many more homology classes, compared to the lower periodicity patterns.

The unit cell with even side length and only 1 defect is an exceptional case. In this case, the homology group is $\mathbb{Z}^2$, where both the cuts correspond to global homology numbers. This is the same as for the unit cell with no defect. Figure 8.4 gives an example of two domino configurations with an oriented transition graph that wraps around the defect. Although this seems to create a non-trivial cycle, it is possible to expand the cycle away from the defect using domino flips, eventually shrinking it to nothing. The domino configurations generated by this unit cell are those with periodicity equal to the spacing between defects. Therefore, the lowest periodicity domino patterns are unhindered by any homology constraint caused by defects.

8.1.2 Odd Spaced Defects

So far, we have considered the unit cell with even side lengths; in other words, the defect spacing is even. For odd-valued spacing between defects, we obtain a different homology group. The 5×5 unit cell with 1 defect is shown in Figure 8.5. Since the unit cell has an odd width and height, it is not possible to consistently colour squares like a checkerboard. As can be seen in Figure 8.5(d), this results in oriented transition graphs that change direction at the periodic boundary of the unit cell. In

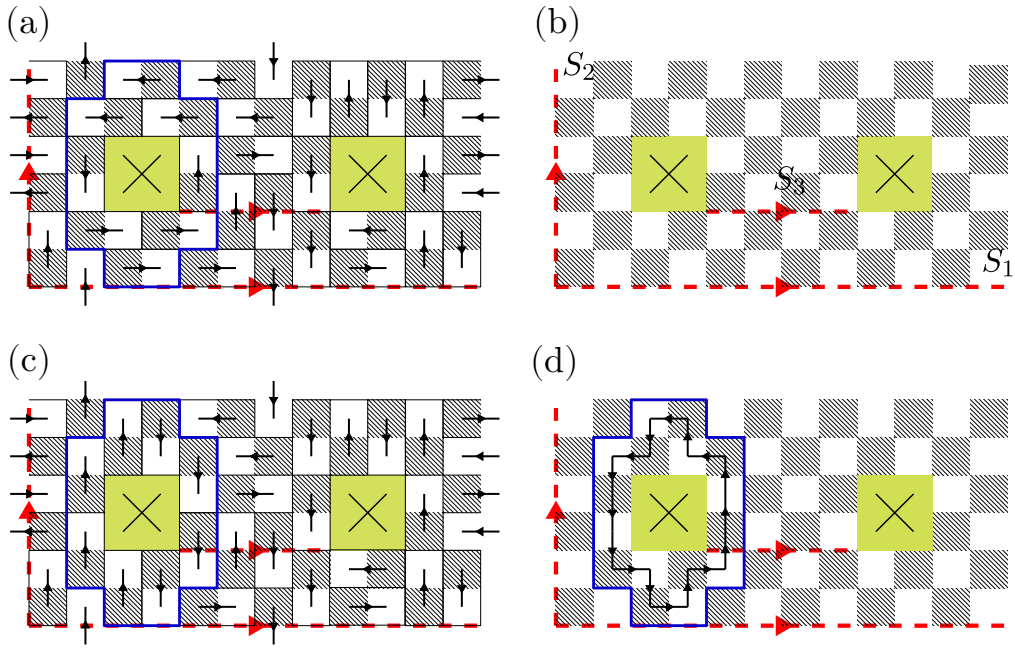


Figure 8.3: (a) and (c) Contain domino configurations on the 12×6 unit cell with 2 2×2 defects. (d) Gives the oriented transition graph resulting from the difference between (a) and (c). In each diagram, a blue highlighted region indicates where dominoes have been rearranged. (b) Shows the cuts of this cell, for reference.

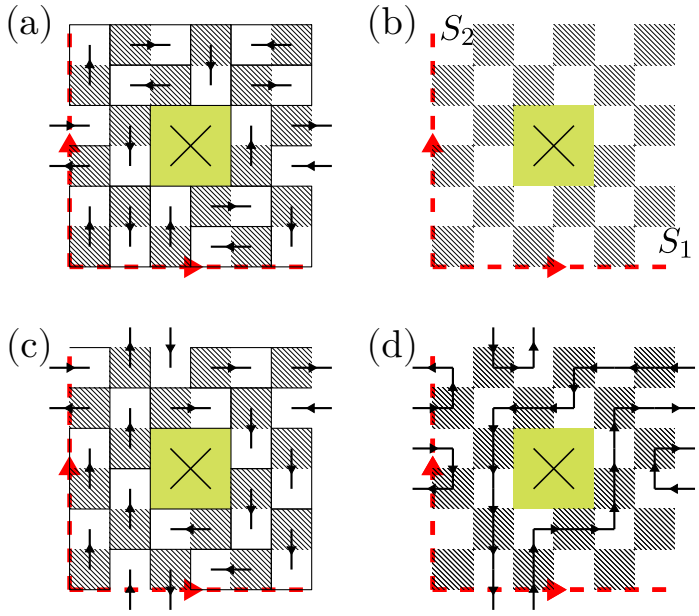


Figure 8.4: (a) and (c) Two domino configurations of the 6×6 unit cell with a 2×2 defect. The cuts used to calculate the homology group are shown in (b), and the oriented transition graph is shown in (d). Although the oriented transition graph contains an anticlockwise cycle around the defect, the homology class is trivial.

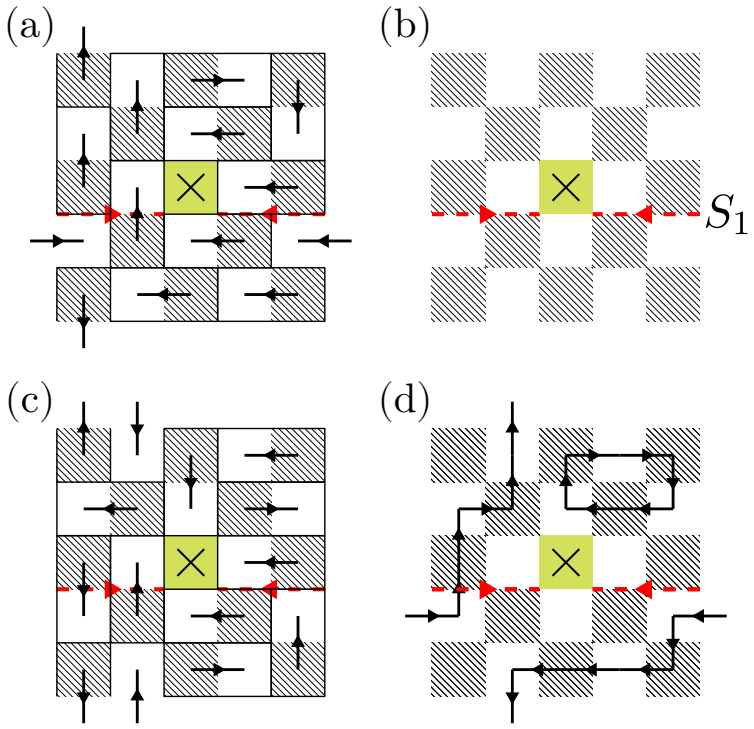


Figure 8.5: 5×5 unit cell with single defect. (a) and (c) Show two domino configurations with oriented transition graph (d) that generates an element in each of the homology classes. (b) The cut that correctly calculates the homology class.

Ref. [43], it is explained that local (or twisted) coefficients can be used for the chain complex, while still giving the correct requirement for domino configurations to be connected via flip moves.

Twisted coefficients allow for nontrivial loops that reverse the orientation of the chain complex. For the domino configurations, these are the loops that pass through an odd number of squares, as stated in Ref. [43]. In the particular case of the unit cell of odd sides with 1 defect, the orientation-reversing loops are those that pass an odd number of times around the periodic boundary in total, either horizontally or vertically. This is entirely analogous to antiperiodic boundary conditions.

To calculate the homology group, we can use a CW complex for the unit cell, as shown in Figure 8.6. The homology group is calculated to be $\mathbb{Z}$, and the cycle $b-a$ can be used as a representative cycle that generates an element in each homology class. The oriented transition graph given in Figure 8.5 is in fact equivalent to the representative cycle for the 5×5 unit cell with single defect.

For this problem, we must use a different cut than before, which is shown in Figure 8.5(b). This cut gives the correct homology class for the representative

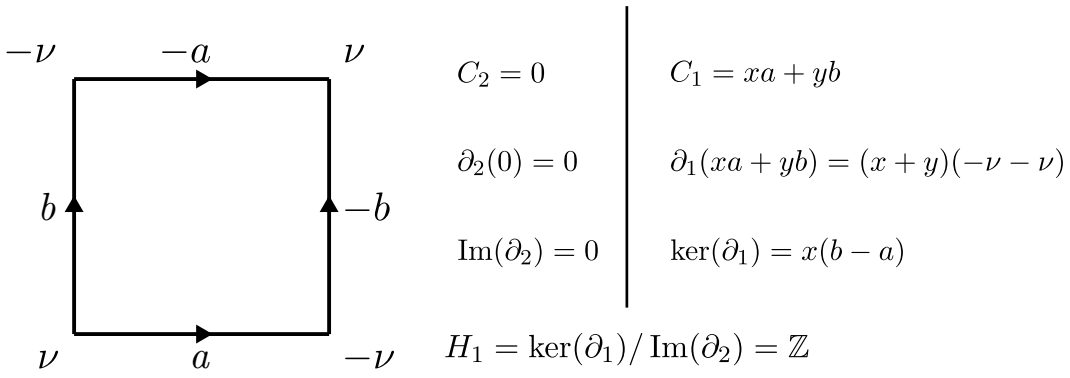


Figure 8.6: CW complex for the deformation retract of the unit cell of odd side length with 1 defect. Coefficients are reversed on either side of the unit cell. The homology group calculation is given to the right.

cycle, and an element of $\mathrm{Im}(\partial_2)$ placed at the periodic boundary will not change the value of the cut because of the property of the orientation to swap on either side of the periodic boundary. Thus, we have a valid method for calculating the homology class of domino configurations on the 5×5 unit cell with 1 defect, although the interpretation of the homology class is not as clear.

The 10×10 unit cell can be used to gain more understanding of the 5×5 unit cell, since the 10×10 unit cell does not have twisted coefficients. The double cover of the 5×5 unit cell with 1 defect gives the 10×10 unit cell with 4 defects. The 10×10 unit cell has homology group $\mathbb{Z}^5$, including 3 defect homology numbers and 2 global homology numbers, as can be seen in Figure 8.7. The cut between the two lower defects is in fact the same as the cut that was used for the 5×5 unit cell with single defect. This means the domino patterns that were not connected by a sequence of domino flips on the 5×5 unit cell remain unconnected.

The oriented transition graph shown for the 10×10 unit cell in Figure 8.7(a) is the same as that for the 5×5 unit cell shown in Figure 8.5. The difference for the 10×10 unit cell is that a checkerboard can be drawn consistently, giving the usual kind of cycle. Looking at the flow across the cuts, it becomes clear that the global homology numbers are zero for this cycle, and the defect homology numbers indicate that the oriented transition graph can be shrunk to the loops shown in Figure 8.7(b).

Therefore, thinking of the 5×5 unit cell with 1 defect as tiling the plane, the possible homology classes are those with multiples of half integer cycles per defect, with no global homology number. We can also say that for defects with odd-valued spacing, the domino patterns with lowest periodicity can have nonzero defect

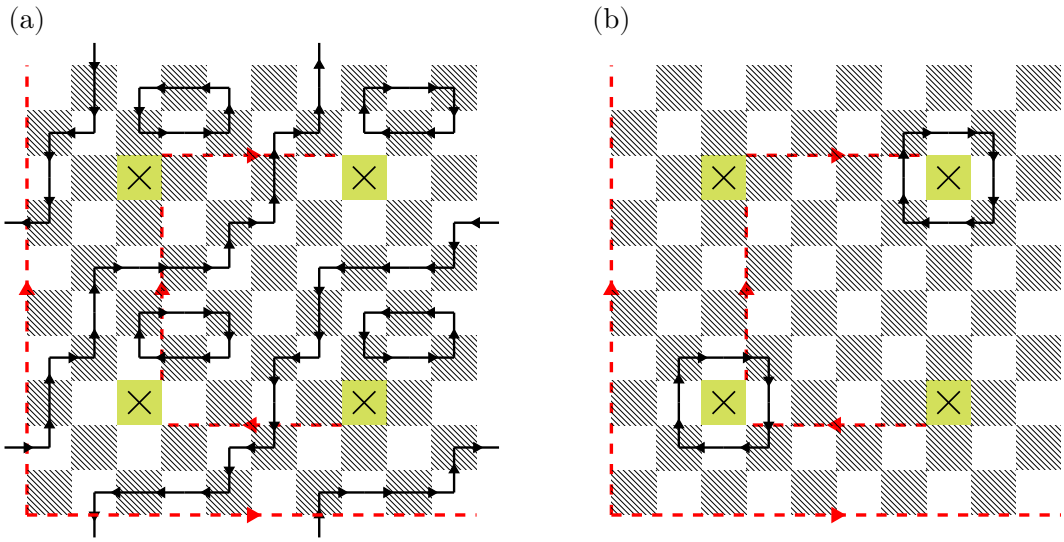


Figure 8.7: (a) Oriented transition graph, the same as in Figure 8.5(d), but here shown in the 10×10 unit cell with 4 defects. (b) An oriented transition graph in the same homology class as in (a), when considering the 5 cuts shown as red dashed arrows, which give the homology group of the 10×10 unit cell with 4 defects.

homology number, but must have zero global homology number. The opposite is true for defects with even-valued spacing.

8.2 Interacting Dominoes And Defects

We have described the effect of defects on the connectivity of domino configurations by flip moves. Now we examine a few of the properties of the set of domino configurations for a specific example, the 7×7 unit cell with 1 defect. Since we have been exploring the intermolecular interactions, a first simplification is to suppose the fixed defects have zero interaction with the dominoes. However, this could be extended by introducing molecule-defect interaction parameters.

In Figure 8.8, we show the convex hull over the set of all configurations of the 7×7 unit cell with some example vertex configurations illustrated. In some cases, the defect acts as an obstacle to achieving an extreme interaction count. For example, small values of n_{ab} and n_{ac} were possible over a range of n_{bc} values for the unit cell without defect; however, here, the least value of n_{ab} and n_{ac} inevitably rises with decreasing n_{bc}. This is because for the unit cell without defect, it is possible to have all dominoes aligned and shift past each other. The defect is an obstruction that causes a change to this boundary element of the convex hull.

Conversely, there are some motifs that are easier to form on the unit cell with

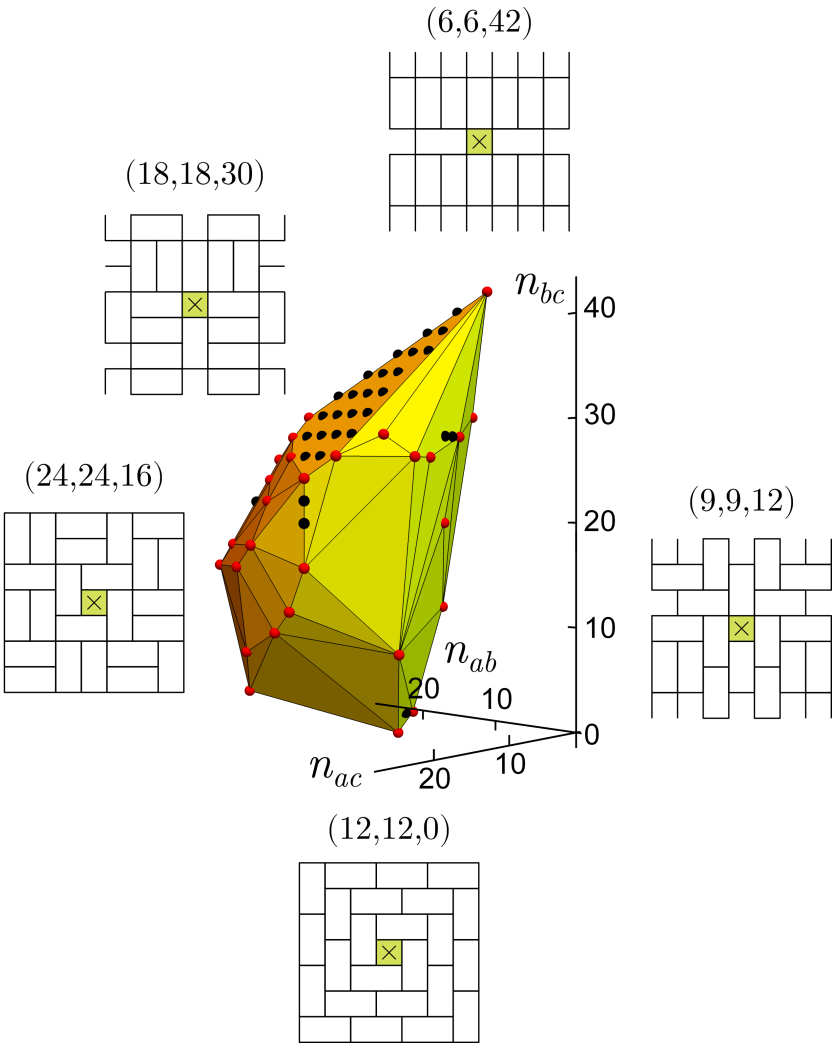


Figure 8.8: The convex hull of points in $\vec{n}$-space is shown corresponding to domino configurations generated by the 7×7 unit cell with 1 defect. Red points are vertex domino configurations and black points are other boundary configurations. 5 Example vertex domino configurations are illustrated along with their $\vec{n}$-vectors.

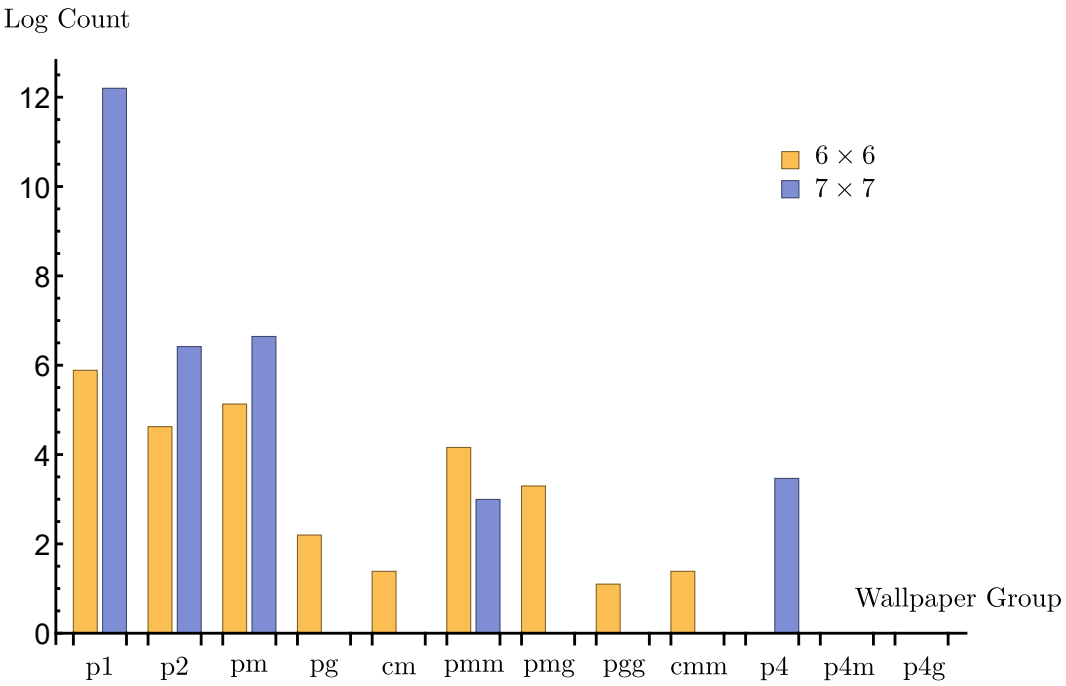


Figure 8.9: Histogram showing Log of the number of domino configurations, versus the wallpaper group. Histograms for both the 6×6 unit cell and the 7×7 unit cell with 1 defect are included.

defect. Most striking are the domino configurations that spiral around the central defect. The $(24, 24, 16)$ and $(12, 12, 0)$ domino configurations shown in Figure 8.8 both have this spiral motif to differing extents. In Figure 8.9, we compare the number of wallpaper groups of the 6×6 unit cell with no defect to that of the 7×7 unit cell with 1 defect. It seems that the two types of unit cell generate quite different fractions of symmetries. The 6×6 unit cell has a more even spread of symmetry groups, while the 7×7 unit cell with defect has peak numbers in the pmm and p4 wallpaper groups. This suggests the introduction of defects can promote specific symmetry as well as changing the connectivity of configurations, both without the defects having any energetic interaction with the dominoes. The introduction of fixed defects provides a further concept for the design principles that we have defined in this thesis.

The homology properties of the 7×7 unit cell are the same as those of the 5×5 unit cell described previously. Both of these unit cells have a single defect homology number (integer) and no global homology number. However, one point that differentiates them is the range of values that the homology number can take. The oriented transition graph between two domino configurations on the same unit

cell consists of loops of value ± 1. Therefore, the homological difference between the configurations is limited by the amount of squares on the unit cell. This also holds true for the even spaced defects.

For the 7×7 unit cell with 1 defect, there are 7 sectors. However, the two most extreme sectors consist of only the two chiral pair domino configurations with strongly spiralling motif shown previously in Figure 8.8, with $\vec{n} = (12, 12, 0)$. More generally, the sectors with greater value have a greater circulation of dominoes around the defects. For the 5 remaining sectors, 5 convex hulls formed by the points in $\vec{n}$-space corresponding to their domino configurations are shown in Figure 8.10. In contrast to the unit cell with no defects, each sector of the 7×7 unit cell with 1 defect contains all symmetry-equivalent domino configurations. On the other hand, for both types of unit cell, the extreme sectors have the fewest domino configurations corresponding to them.

In summary, defects further split the connectivity of domino configurations under local rearrangements. Furthermore, this splitting depends on the spacing between defects, the periodicity of configurations, and the number of dominoes that can simultaneously rearrange. Even inert defects can promote symmetry and alter the landscape of lowest energy states; that is, the boundary components of convex hulls corresponding to sets of kinetically accessible configurations.

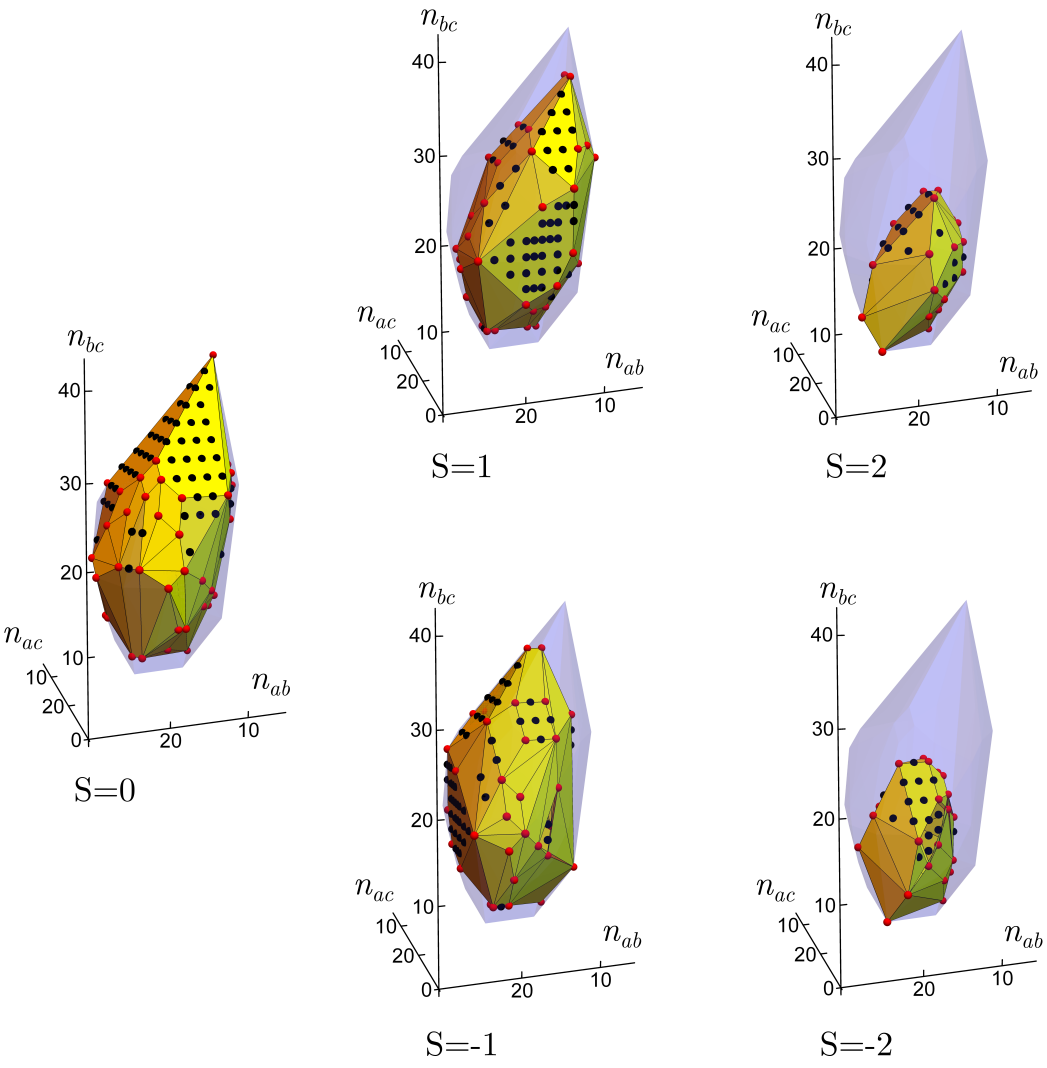


Figure 8.10: For the 7×7 unit cell with single defect, convex hulls are shown for each of the subsets formed of domino configurations with the same value of their sector, S. A blue shadow is shown to indicate the convex hull of the combined set of points.

Chapter 9

Conclusion

In this thesis we have analysed molecular monolayers using a fully-packed polyomino model with nearest-neighbour interactions. Furthermore, we describe design principles that prescribe the changes to the parameters of a molecular system that would favour a new molecular network with desired properties. This level of experimental control has been demonstrated in the literature, providing exciting opportunities for the design of molecular networks. In particular, several experiments have shown that the directionality of molecular interactions, effective packing shape, and temperature can be adjusted to give rise to new molecular networks. For example, the bromine substituents of pyrene derivatives described in Ref. [8] are modified to alter the directional interactions between molecules, resulting in different molecular networks. In Ref. [27], heterocirculenes with 3 different shapes are chosen, each forming different molecular networks due to the packing shape and van der Waals interactions between molecular units. Furthermore, the work in Ref. [10] shows that molecular networks equivalent to uniform random tilings are possible, where the experimental conditions can be altered to change the effective interactions, causing a transition between ordered and random molecular networks.

For the analysis of 2D supramolecular networks in this thesis, we have outlined several methods and concepts that were used specifically to investigate the fully packed periodic molecular patterns. First is our algorithm for enumerating polyomino configurations on the square lattice with periodic boundary conditions, using algorithm DLX. Second is our symmetry-finding algorithm that makes use of the restricted set of possible symmetries of polyomino configurations, allowing us to greatly reduce the search space of symmetries. We discovered a bottleneck in our symmetry algorithm, so we build on the idea of exploring discrete symmetries by creating a fast algorithm that uses group theory concepts to determine whether

there are any extra symmetries. We also found a surprisingly simple method to account for polyomino configurations that are equivalent as planar patterns, using the symmetry properties of the configuration.

An important part of our investigation into polyomino patterns is to leave the interaction strengths as free parameters, making it possible to find the combination of interactions that favour a given property of the molecular tiling. We first implemented a differential evolution algorithm to optimise the interaction parameters over a test fitness function - the mean rotational symmetry of the ensemble of configurations. We found that the solutions at infinite parameter values - corresponding to low temperature - were important for this problem. We took this as an indication to create an algorithm that explicitly looks for these low temperature solutions. Therefore, we switched to a convex hull construction that compares the polyomino configurations within the space of interaction counts - $\vec{n}$-space. This convex hull construction allows for the possible ground states to be viewed as boundary components of a convex hull in $\vec{n}$-space, giving a simple yet useful way to identify the low temperature solutions.

The first polyomino system we analysed is the domino system. We found that the vertex configurations corresponding to the possible unique lowest energy patterns are often highly symmetric, and quite limited in number, showing that the fraction of patterns that can be designed as lowest energy states is very small, even though there are several interaction parameters that take on any continuous value. As well as identifying lowest energy states, $\vec{n}$-space is useful in visualising several of the quantities of domino patterns, for example the degeneracy, which increases going toward the centre of points. Also, the chirality of domino patterns can be easily interpreted in the $\vec{n}$ space, since only the plane $n_{ab} = n_{ac}$ contains achiral patterns. Thus, interactions perpendicular to this plane strongly favour one chirality over the other. We found that low energy states corresponding to edges and faces of the convex hull require finely tuned interaction parameters and that in some cases the ground state can be highly degenerate, or correspond to particular sets of domino patterns; for example, the patterns with dominoes of only one orientation lie on the line $n_{ab} = n_{ac} = 0$.

Using our algorithm for generating dominoes, we found that it is possible to enumerate up to a given primitive unit cell size, which allows for the natural interpretation of enumerating periodic patterns in order of periodicity. Furthermore, by including all domino patterns with primitive cell area 8 and below, we achieve a limit shape for the convex hull, which gives the possible low energy states over the set of all possible planar domino patterns. By splitting the domino configurations

according to their connectivity via local rearrangements, we found new convex hulls for the subsets, allowing for effective lowest energy states that are disconnected from the true lowest energy state due to the lack of domino rearrangement sequence between them. We also investigate heat capacity curves over multiple directions for the vector of interaction parameters, and we found that the transition to the high-entropy state occurs at different temperatures, depending on the relative values of interaction parameters.

For polyominoes, we generalise many of the concepts that were described for dominoes, such as high degeneracy ground states, connectivity by rearrangements, and components of the convex hull boundary that form interesting subsets. We also found that a reduced set of interactions can be useful in describing polyomino systems that would otherwise have a prohibitively large number of interactions. In contrast to the domino system, we found that for straight trominoes, configurations permit shifted straight trominoes at many different places, giving an easy axis in $\vec{n}$-space. For T-tetrominoes, strong entropic forces give the impression of a pseudo-constraint on the unit cell.

Due to the computational difficulty of enumerating large unit cells, we designed an extension of algorithm DLX that can be used to sample from solutions of the exact cover problem with uniform probability. We tested this algorithm for our domino system and found it works well given a suitable value for the parameter Q that controls the frequency of backward transitions towards the root of the tree of partial solutions. We used this random DLX algorithm to compare a theoretical large unit cell limit of the degeneracy peak with the sampled peaks for finite unit cells, finding that the values of the peak do not coincide for the unit cells we tested. In particular, the degeneracy in the n_{bc} component is slowest to converge. However, the convergence to the limit for larger unit cells is not infeasible.

Finally, we made an extension by including fixed defects in the unit cell, and we found that homology techniques could again be used to calculate the subsets of domino configurations that are connected by sequences of domino flip moves. We found that the defects further split the subsets of connected domino configurations and promote configurations with 4-fold rotational symmetry. Furthermore, we found that the connected subsets depend on the spacing of defects and the periodicity of configurations. This indicates that fixed defects without an explicit interaction with the dominoes could be used for stabilising desired domino patterns.

There are several possible avenues for future work, we would like to mention the two most interesting ones. The random DLX algorithm could potentially be used on more general exact cover problems and could be made more efficient by choosing

backwards rates that differ at each level of the search tree. Also, it could benefit from a further acceptance/rejection criterion to find important configurations. In another direction, the behaviour of the molecular system at high packing fraction but with a very small number of mobile vacancies could give more understanding of the point between full packing and lower density. In particular, the diffusion of mobile vacancies is known to be suppressed by aligning interactions between dominoes [50]. Therefore, if large fixed defects are introduced to such a system, the mobile vacancies may be unable to change the effective homology class, giving an effective homological constraint that is eroded with increasing temperature.

Bibliography

[1] Shengbin Lei, Kazukuni Tahara, Frans C. De Schryver, Mark Van der Auweraer, Yoshito Tobe, and Steven De Feyter. One building block, two different supramolecular surface-confined patterns: Concentration in control at the solid–liquid interface. *Angewandte Chemie International Edition*, 47(16): 2964–2968, apr 2008. doi: 10.1002/anie.200705322. URL https://doi.org/10.1002%2Fanie.200705322.

[2] Yi Hu, Kai Miao, Bao Zha, Li Xu, Xinrui Miao, and Wenli Deng. Fabrication of chiral networks for a tri-substituted anthraquinone derivative using molecular self-assembly. *Phys. Chem. Chem. Phys.*, 18(19):13164–13168, 2016. doi: 10.1039/c6cp02070d. URL http://dx.doi.org/10.1039/c6cp02070d.

[3] Kazukuni Tahara, Shuhei Furukawa, Hiroshi Uji-i, Tsutomu Uchino, Tomoyuki Ichikawa, Jian Zhang, Wael Mamdouh, Motohiro Sonoda, Frans C. De Schryver, Steven De Feyter, and Yoshito Tobe. Two-dimensional porous molecular networks of dehydrobenzo[12]annulene derivatives via alkyl chain interdigitation. *Journal of the American Chemical Society*, 128(51):16613–16625, dec 2006. doi: 10.1021/ja0655441. URL https://doi.org/10.1021%2Fja0655441.

[4] Stephen Whitelam. Examples of molecular self-assembly at surfaces. *Adv. Mater.*, 27(38):5720–5725, apr 2015. doi: 10.1002/adma.201405573. URL http://dx.doi.org/10.1002/adma.201405573.

[5] Alex Saywell, Graziano Magnano, Christopher J. Satterley, Luís M.A. Perdigão, Andrew J. Britton, Nassiba Taleb, María del Carmen Giménez-López, Neil R. Champness, James N. O'Shea, and Peter H. Beton. Self-assembled aggregates formed by single-molecule magnets on a gold surface. *Nature Communications*, 1(6):1–8, sep 2010. doi: 10.1038/ncomms1075. URL https://doi.org/10.1038%2Fncomms1075.

[6] Anna G. Slater, Peter H. Beton, and Neil R. Champness. Two-dimensional supramolecular chemistry on surfaces. *Chem. Sci.*, 2(8):1440, 2011. doi: 10.1039/c1sc00251a. URL http://dx.doi.org/10.1039/C1SC00251A.

[7] U. K. Weber, V. M. Burlakov, L. M. A. Perdigão, R. H. J. Fawcett, P. H. Beton, N. R. Champness, J. H. Jefferson, G. A. D. Briggs, and D. G. Pettifor. Role of interaction anisotropy in the formation and stability of molecular templates. *Physical Review Letters*, 100(15), apr 2008. doi: 10.1103/physrevlett.100.156101. URL https://doi.org/10.1103%2Fphysrevlett.100.156101.

[8] Tuan Anh Pham, Fei Song, Manh-Thuong Nguyen, and Meike Stöhr. Self-assembly of pyrene derivatives on au(111): substituent effects on intermolecular interactions. *Chem. Commun.*, 50(91):14089–14092, may 2014. doi: 10.1039/c4cc02753a. URL http://dx.doi.org/10.1039/C4CC02753A.

[9] Thomas K. Haxton, Hui Zhou, Isaac Tamblyn, Daejin Eom, Zonghai Hu, Jeffrey B. Neaton, Tony F. Heinz, and Stephen Whitelam. Competing thermodynamic and dynamic factors select molecular assemblies on a gold surface. *Phys. Rev. Lett.*, 111(26):265701, dec 2013. doi: 10.1103/physrevlett.111.265701. URL http://dx.doi.org/10.1103/PhysRevLett.111.265701.

[10] Andrew Stannard, James C. Russell, Matthew O. Blunt, Christos Salesiotis, María del Carmen Giménez-López, Nassiba Taleb, Martin Schröder, Neil R. Champness, Juan P. Garrahan, and Peter H. Beton. Broken symmetry and the variation of critical properties in the phase behaviour of supramolecular rhombus tilings. *Nat. Chem.*, 4(2):112–117, nov 2011. doi: 10.1038/nchem.1199. URL http://dx.doi.org/10.1038/nchem.1199.

[11] G. M. Whitesides. Self-assembly at all scales. *Science*, 295(5564):2418–2421, mar 2002. doi: 10.1126/science.1070821. URL http://dx.doi.org/10.1126/science.1070821.

[12] Matthew O. Blunt, James C. Russell, Neil R. Champness, and Peter H. Beton. Templating molecular adsorption using a covalent organic framework. *Chem. Commun.*, 46(38):7157, 2010. doi: 10.1039/c0cc01810d. URL http://dx.doi.org/10.1039/C0CC01810D.

[13] N. Bowden. Self-assembly of mesoscale objects into ordered two-dimensional arrays. *Science*, 276(5310):233–235, apr 1997. doi:

10.1126/science.276.5310.233. URL http://dx.doi.org/10.1126/science.276.5310.233.

[14] Marco Bieri, Manh-Thuong Nguyen, Oliver Gröning, Jinming Cai, Matthias Treier, Kamel Aït-Mansour, Pascal Ruffieux, Carlo A. Pignedoli, Daniele Passerone, Marcel Kastler, Klaus Müllen, and Roman Fasel. Two-dimensional polymer formation on surfaces: Insight into the roles of precursor mobility and reactivity. *J. Am. Chem. Soc.*, 132(46):16669–16676, nov 2010. doi: 10.1021/ja107947z. URL http://dx.doi.org/10.1021/ja107947z.

[15] Wenyan Liu, Hong Zhong, Risheng Wang, and Nadrian C. Seeman. Crystalline two-dimensional DNA-origami arrays. *Angew. Chem. Int. Ed.*, 50(1):264–267, nov 2010. doi: 10.1002/anie.201005911. URL http://dx.doi.org/10.1002/anie.201005911.

[16] Bryan Wei, Mingjie Dai, and Peng Yin. Complex shapes self-assembled from single-stranded DNA tiles. *Nature*, 485(7400):623–626, may 2012. doi: 10.1038/nature11075. URL http://dx.doi.org/10.1038/nature11075.

[17] Brian C. Barnes, Daniel W. Siderius, and Lev D. Gelb. Structure, thermodynamics, and solubility in tetromino fluids. *Langmuir*, 25(12):6702–6716, jun 2009. doi: 10.1021/la900196b. URL http://dx.doi.org/10.1021/la900196b.

[18] S. E. Ahnert, I. G. Johnston, T. M. A. Fink, J. P. K. Doye, and A. A. Louis. Self-assembly, modularity, and physical complexity. *Phys. Rev. E*, 82(2):026117, aug 2010. doi: 10.1103/physreve.82.026117. URL http://dx.doi.org/10.1103/PhysRevE.82.026117.

[19] Marjolein N. van der Linden, Jonathan P. K. Doye, and Ard A. Louis. Formation of dodecagonal quasicrystals in two-dimensional systems of patchy particles. *J. Chem. Phys.*, 136(5):054904, 2012. doi: 10.1063/1.3679653. URL http://dx.doi.org/10.1063/1.3679653.

[20] Simone Dussi, Lorenzo Rovigatti, and Francesco Sciortino. On the gas–liquid phase separation and the self-assembly of charged soft dumbbells. *Mol. Phys.*, 111(22-23):3608–3617, dec 2013. doi: 10.1080/00268976.2013.838315. URL http://dx.doi.org/10.1080/00268976.2013.838315.

[21] John M. A. Grime, James F. Dama, Barbie K. Ganser-Pornillos, Cora L. Woodward, Grant J. Jensen, Mark Yeager, and Gregory A. Voth. Coarse-grained simulation reveals key features of HIV-1 capsid self-assembly. *Nat.*

Commun., 7:11568, may 2016. doi: 10.1038/ncomms11568. URL http://dx.doi.org/10.1038/ncomms11568.

[22] Timothy R. Cook, Yao-Rong Zheng, and Peter J. Stang. Metal–organic frameworks and self-assembled supramolecular coordination complexes: Comparing and contrasting the design, synthesis, and functionality of metal–organic materials. *Chem. Rev.*, 113(1):734–777, jan 2013. doi: 10.1021/cr3002824. URL http://dx.doi.org/10.1021/cr3002824.

[23] Xuemei Zhang, Qingdao Zeng, and Chen Wang. On-surface single molecule synthesis chemistry: a promising bottom-up approach towards functional surfaces. *Nanoscale*, 5(18):8269, 2013. doi: 10.1039/c3nr01611k. URL https://doi.org/10.1039%2Fc3nr01611k.

[24] Lei Dong, Zi'Ang Gao, and Nian Lin. Self-assembly of metal–organic coordination structures on surfaces. *Progress in Surface Science*, 91(3):101–135, aug 2016. doi: 10.1016/j.progsurf.2016.08.001. URL https://doi.org/10.1016%2Fj.progsurf.2016.08.001.

[25] Jun Liu, Tao Lin, Ziliang Shi, Fei Xia, Lei Dong, Pei Nian Liu, and Nian Lin. Structural transformation of two-dimensional metal–organic coordination networks driven by intrinsic in-plane compression. *Journal of the American Chemical Society*, 133(46):18760–18766, nov 2011. doi: 10.1021/ja2056193. URL https://doi.org/10.1021%2Fja2056193.

[26] Kazukuni Tahara, Tanya Balandina, Shuhei Furukawa, Steven De Feyter, and Yoshito Tobe. Molecular pentagonal tiling: self-assemblies of pentagonal-shaped macrocycles at liquid/solid interfaces. *CrystEngComm*, 13(18): 5551, 2011. doi: 10.1039/c1ce05336a. URL https://doi.org/10.1039%2Fc1ce05336a.

[27] W. D. Xiao, Y. Y. Zhang, L. Tao, K. At-Mansour, K. Y. Chernichenko, V. G. Nenajdenko, P. Ruffieux, S. X. Du, H.-J. Gao, and R. Fasel. Impact of heterocirculene molecular symmetry upon two-dimensional crystallization. *Scientific Reports*, 4(1), jun 2014. doi: 10.1038/srep05415. URL https://doi.org/10.1038%2Fsrep05415.

[28] Natalia Martsinovich and Alessandro Troisi. Modeling the self-assembly of benzenedicarboxylic acids using monte carlo and molecular dynamics simulations. *J. Phys. Chem. C*, 114(10):4376–4388, mar 2010. doi: 10.1021/jp911671b. URL http://dx.doi.org/10.1021/jp911671b.

[29] Eric Jankowski and Sharon C. Glotzer. Calculation of partition functions for the self-assembly of patchy particles. *J. Phys. Chem. B*, 115(48):14321–14326, dec 2011. doi: 10.1021/jp206430z. URL http://dx.doi.org/10.1021/jp206430z.

[30] Trung Dac Nguyen, Eric Jankowski, and Sharon C. Glotzer. Self-assembly and reconfigurability of shape-shifting particles. *ACS Nano*, 5(11):8892–8903, nov 2011. doi: 10.1021/nn203067y. URL http://dx.doi.org/10.1021/nn203067y.

[31] Michael Korn and Igor Pak. Tilings of rectangles with t-tetrominoes. *Theor. Comput. Sci.*, 319(1-3):3–27, jun 2004. doi: 10.1016/j.tcs.2004.02.023. URL http://dx.doi.org/10.1016/j.tcs.2004.02.023.

[32] Joyjit Kundu, R. Rajesh, Deepak Dhar, and Jürgen F. Stilck. Nematic-disordered phase transition in systems of long rigid rods on two-dimensional lattices. *Phys. Rev. E*, 87(3):032103, mar 2013. doi: 10.1103/physreve.87.032103. URL http://dx.doi.org/10.1103/PhysRevE.87.032103.

[33] Richard Kenyon. Lectures on Dimers. *ArXiv e-prints*, October 2009. URL https://arxiv.org/abs/0910.3129.

[34] Michael E. Fisher. Statistical mechanics of dimers on a plane lattice. *Phys. Rev.*, 124(6):1664–1672, dec 1961. doi: 10.1103/physrev.124.1664. URL http://dx.doi.org/10.1103/PhysRev.124.1664.

[35] P. W. Kasteleyn. The statistics of dimers on a lattice. *Physica*, 27(12):1209–1225, dec 1961. doi: 10.1016/0031-8914(61)90063-5. URL http://dx.doi.org/10.1016/0031-8914(61)90063-5.

[36] P. W. Kasteleyn. Dimer statistics and phase transitions. *J. Math. Phys.*, 4 (2):287, 1963. doi: 10.1063/1.1703953. URL http://dx.doi.org/10.1063/1.1703953.

[37] Daniel S. Rokhsar and Steven A. Kivelson. Superconductivity and the quantum hard-core dimer gas. *Phys. Rev. Lett.*, 61(20):2376–2379, nov 1988. doi: 10.1103/physrevlett.61.2376. URL http://dx.doi.org/10.1103/PhysRevLett.61.2376.

[38] R. Moessner and S. L. Sondhi. Resonating valence bond phase in the triangular lattice quantum dimer model. *Phys. Rev. Lett.*, 86(9):1881–1884, feb

2001. doi: 10.1103/physrevlett.86.1881. URL http://dx.doi.org/10.1103/PhysRevLett.86.1881.

[39] Joyjit Kundu and R. Rajesh. Phase transitions in a system of hard rectangles on the square lattice. *Phys. Rev. E*, 89(5):052124, may 2014. doi: 10.1103/physreve.89.052124. URL http://dx.doi.org/10.1103/PhysRevE.89.052124.

[40] Kabir Ramola, Kedar Damle, and Deepak Dhar. Columnar order and ashkin-teller criticality in mixtures of hard squares and dimers. *Phys. Rev. Lett.*, 114(19):190601, may 2015. doi: 10.1103/physrevlett.114.190601. URL http://dx.doi.org/10.1103/PhysRevLett.114.190601.

[41] Christopher L. Henley. Relaxation time for a dimer covering with height representation. *J. Stat. Phys.*, 89(3-4):483–507, nov 1997. doi: 10.1007/bf02765532. URL http://dx.doi.org/10.1007/BF02765532.

[42] Henry Cohn, Richard Kenyon, and James Propp. A variational principle for domino tilings. *J. Am. Math. Soc*, 14(02):297–347, apr 2001. doi: 10.1090/s0894-0347-00-00355-6. URL http://dx.doi.org/10.1090/S0894-0347-00-00355-6.

[43] N. C. Saldanha, C. Tomei, M. A. Casarin, and D. Romualdo. Spaces of domino tilings. *Discrete Comput. Geom.*, 14(2):207–233, sep 1995. doi: 10.1007/bf02570703. URL http://dx.doi.org/10.1007/BF02570703.

[44] Eric Rémila. The lattice structure of the set of domino tilings of a polygon. *Theoretical Computer Science*, 322(2):409–422, aug 2004. doi: 10.1016/j.tcs.2004.03.020. URL https://doi.org/10.1016%2Fj.tcs.2004.03.020.

[45] Sébastien Desreux and Eric Rémila. An optimal algorithm to generate tilings. *Journal of Discrete Algorithms*, 4(1):168–180, mar 2006. doi: 10.1016/j.jda.2005.01.003. URL https://doi.org/10.1016%2Fj.jda.2005.01.003.

[46] James Gary Propp and David Bruce Wilson. Exact sampling with coupled markov chains and applications to statistical mechanics. *Random Structures and Algorithms*, 9(1-2):223–252, aug 1996. doi: 10.1002/(SICI)1098-2418(199608/09)9:1/2⟨223::AID-RSA14⟩3.0.CO;2-O. URL http://onlinelibrary.wiley.com/doi/10.1002/(SICI)1098-2418(199608/09)9:1/2%3C223::AID-RSA14%3E3.0.CO;2-O/abstract.

[47] Michael E. Fisher and John Stephenson. Statistical mechanics of dimers on a plane lattice. II. dimer correlations and monomers. *Physical Review*, 132(4):

1411–1431, nov 1963. doi: 10.1103/physrev.132.1411. URL https://doi.org/10.1103%2Fphysrev.132.1411.

[48] Sébastien Desreux, Martin Matamala, Ivan Rapaport, and Eric Rémila. Domino tilings and related models: space of configurations of domains with holes. *Theoretical Computer Science*, 319(1-3):83–101, jun 2004. doi: 10.1016/j.tcs.2004.02.020. URL https://doi.org/10.1016%2Fj.tcs.2004.02.020.

[49] Fabien Alet, Jesper Lykke Jacobsen, Grégoire Misguich, Vincent Pasquier, Frédéric Mila, and Matthias Troyer. Interacting classical dimers on the square lattice. *Phys. Rev. Lett.*, 94(23):235702, jun 2005. doi: 10.1103/physrevlett.94.235702. URL http://dx.doi.org/10.1103/PhysRevLett.94.235702.

[50] Fabien Alet, Yacine Ikhlef, Jesper Lykke Jacobsen, Grégoire Misguich, and Vincent Pasquier. Classical dimers with aligning interactions on the square lattice. *Physical Review E*, 74(4), oct 2006. doi: 10.1103/physreve.74.041124. URL https://doi.org/10.1103%2Fphysreve.74.041124.

[51] Richard Kenyon. Dominos and the gaussian free field. *The Annals of Probability*, 29(3):1128–1137, jul 2001. doi: 10.1214/aop/1015345599. URL https://doi.org/10.1214%2Faop%2F1015345599.

[52] Alessandro Giuliani, Vieri Mastropietro, and Fabio Lucio Toninelli. Height fluctuations in interacting dimers. *Annales de l'Institut Henri Poincaré, Probabilités et Statistiques*, 53(1):98–168, feb 2017. doi: 10.1214/15-aihp710. URL https://doi.org/10.1214%2F15-aihp710.

[53] Stefanos Papanikolaou, Daniel Charrier, and Eduardo Fradkin. Ising nematic fluid phase of hard-core dimers on the square lattice. *Physical Review B*, 89(3), jan 2014. doi: 10.1103/physrevb.89.035128. URL https://doi.org/10.1103%2Fphysrevb.89.035128.

[54] Hiromi Otsuka. Classical dimer model with anisotropic interactions on the square lattice. *Phys. Rev. E*, 80(1):011140, jul 2009. doi: 10.1103/physreve.80.011140. URL http://dx.doi.org/10.1103/PhysRevE.80.011140.

[55] David A. Huse, Werner Krauth, R. Moessner, and S. L. Sondhi. Coulomb and liquid dimer models in three dimensions. *Physical Review Letters*, 91(16), oct 2003. doi: 10.1103/physrevlett.91.167004. URL https://doi.org/10.1103%2Fphysrevlett.91.167004.

[56] Fabien Alet, Grégoire Misguich, Vincent Pasquier, Roderich Moessner, and Jesper Lykke Jacobsen. Unconventional continuous phase transition in a three-dimensional dimer model. *Physical Review Letters*, 97(3), jul 2006. doi: 10.1103/physrevlett.97.030403. URL https://doi.org/10.1103%2Fphysrevlett.97.030403.

[57] S. W. Golomb. Checker boards and polyominoes. *The American Mathematical Monthly*, 61(10):675, dec 1954. doi: 10.2307/2307321. URL https://doi.org/10.2307%2F2307321.

[58] Aleksandra Woszczyk and Paweł Szabelski. Theoretical investigations of the 2d chiral segregation induced by external directional fields. *RSC Adv.*, 5 (100):81933–81942, 2015. doi: 10.1039/c5ra15192a. URL http://dx.doi.org/10.1039/C5RA15192A.

[59] Cristopher Moore. Some polyomino tilings of the plane. 1999. URL https://arxiv.org/abs/math/9905012v1.

[60] Anandamohan Ghosh, Deepak Dhar, and Jesper L. Jacobsen. Random trimer tilings. *Phys. Rev. E*, 75(1):011115, jan 2007. doi: 10.1103/physreve.75.011115. URL http://dx.doi.org/10.1103/PhysRevE.75.011115.

[61] Jurijs Kirillovs. Polyomino coloring and complex numbers. *Theoretical Computer Science*, 400(1-3):100–112, jun 2008. doi: 10.1016/j.tcs.2008.02.033. URL https://doi.org/10.1016%2Fj.tcs.2008.02.033.

[62] Igor Pak. Ribbon tile invariants. *Transactions of the American Mathematical Society*, 352(12):5525–5562, dec 2000. doi: 10.1090/s0002-9947-00-02666-0. URL https://doi.org/10.1090%2Fs0002-9947-00-02666-0.

[63] Werner Krauth and R. Moessner. Pocket monte carlo algorithm for classical doped dimer models. *Physical Review B*, 67(6), feb 2003. doi: 10.1103/physrevb.67.064503. URL https://doi.org/10.1103%2Fphysrevb.67.064503.

[64] J.H Conway and J.C Lagarias. Tiling with polyominoes and combinatorial group theory. *Journal of Combinatorial Theory, Series A*, 53(2):183–208, mar 1990. doi: 10.1016/0097-3165(90)90057-4. URL https://doi.org/10.1016%2F0097-3165%2890%2990057-4.

[65] Michael Reid. Tile homotopy groups, 2003. URL https://doi.org/10.5169/seals-66684.

[66] Igor Pak. Tile invariants: new horizons. *Theoretical Computer Science*, 303 (2-3):303–331, jul 2003. doi: 10.1016/s0304-3975(02)00495-4. URL https://doi.org/10.1016%2Fs0304-3975%2802%2900495-4.

[67] D. W. Walkup. Covering a rectangle with t-tetrominoes. *The American Mathematical Monthly*, 72(9):986, nov 1965. doi: 10.2307/2313337. URL https://doi.org/10.2307%2F2313337.

[68] Jesper Lykke Jacobsen. Tetromino tilings and the tutte polynomial. *Journal of Physics A: Mathematical and Theoretical*, 40(7):1439–1446, jan 2007. doi: 10.1088/1751-8113/40/7/002. URL https://doi.org/10.1088%2F1751-8113%2F40%2F7%2F002.

[69] Klaus Froböse, Frank Bonnemeier, and Josef Jäckle. Orientational correlations in the lattice gas of angular trimers on the square lattice. *Journal of Physics A: Mathematical and General*, 29(3):485–496, feb 1996. doi: 10.1088/0305-4470/29/3/005. URL https://doi.org/10.1088%2F0305-4470%2F29%2F3%2F005.

[70] Solomon W. Golomb. Tiling with sets of polyominoes. *Journal of Combinatorial Theory*, 9(1):60–71, jul 1970. doi: 10.1016/s0021-9800(70)80055-2. URL https://doi.org/10.1016%2Fs0021-9800%2870%2980055-2.

[71] Nicolas Ollinger. Tiling the plane with a fixed number of polyominoes. In *Language and Automata Theory and Applications*, pages 638–647. Springer Berlin Heidelberg, 2009. doi: 10.1007/978-3-642-00982-2_54. URL https://doi.org/10.1007%2F978-3-642-00982-2_54.

[72] H.A.G. Wijshoff and J. van Leeuwen. Arbitrary versus periodic storage schemes and tessellations of the plane using one type of polyomino. *Information and Control*, 62(1):1–25, jul 1984. doi: 10.1016/s0019-9958(84)80007-8. URL https://doi.org/10.1016%2Fs0019-9958%2884%2980007-8.

[73] D. Beauquier and M. Nivat. On translating one polyomino to tile the plane. *Discrete & Computational Geometry*, 6(4):575–592, dec 1991. doi: 10.1007/bf02574705. URL https://doi.org/10.1007%2Fbf02574705.

[74] K. Keating and A. Vince. Isohedral polyomino tiling of the plane. *Discrete & Computational Geometry*, 21(4):615–630, jun 1999. doi: 10.1007/pl00009442. URL https://doi.org/10.1007%2Fpl00009442.

[75] C. Moore and J.M. Robson. Hard tiling problems with simple tiles. *Discrete & Computational Geometry*, 26(4):573–590, jan 2001. doi: 10.1007/s00454-001-0047-6. URL https://doi.org/10.1007%2Fs00454-001-0047-6.

[76] Iwan Jensen and Anthony J Guttmann. Statistics of lattice animals (polyominoes) and polygons. *Journal of Physics A: Mathematical and General*, 33(29):L257–L263, jul 2000. doi: 10.1088/0305-4470/33/29/102. URL https://doi.org/10.1088%2F0305-4470%2F33%2F29%2F102.

[77] D.Hugh Redelmeier. Counting polyominoes: Yet another attack. *Discrete Mathematics*, 36(2):191–203, 1981. doi: 10.1016/0012-365x(81)90237-5. URL https://doi.org/10.1016%2F0012-365x%2881%2990237-5.

[78] Nicolas Allegra. Exact solution of the 2d dimer model: Corner free energy, correlation functions and combinatorics. *Nuclear Physics B*, 894:685–732, may 2015. doi: 10.1016/j.nuclphysb.2015.03.022. URL https://doi.org/10.1016%2Fj.nuclphysb.2015.03.022.

[79] Solomon W. Golomb. Tiling with polyominoes. *Journal of Combinatorial Theory*, 1(2):280–296, sep 1966. doi: 10.1016/s0021-9800(66)80033-9. URL https://doi.org/10.1016%2Fs0021-9800%2866%2980033-9.

[80] Weidong Dou, Dandan Guan, Hongying Mao, Fei Song, Han Huang, Hanjie Zhang, Haiyang Li, Pimo He, and Shining Bao. Aggregation of organic molecules on silver surface with the balance between molecule–substrate interaction and intermolecular interaction. *Chemical Physics Letters*, 470 (1-3):126–130, feb 2009. doi: 10.1016/j.cplett.2009.01.040. URL https://doi.org/10.1016%2Fj.cplett.2009.01.040.

[81] M. Hamermesh. *Group Theory and Its Application to Physical Problems*. Addison Wesley Series in Physics. Dover Publications, 1962. ISBN 9780486661810. URL http://store.doverpublications.com/0486661814.html.

[82] Howard Hiller. Crystallography and cohomology of groups. *The American Mathematical Monthly*, 93(10):765, dec 1986. doi: 10.2307/2322930. URL https://doi.org/10.2307/2322930.

[83] Doris Schattschneider. The plane symmetry groups: Their recognition and notation. *Am. Math. Monthly*, 85(6):439, jun 1978. doi: 10.2307/2320063. URL http://dx.doi.org/10.2307/2320063.

[84] Sebastian Stepanow, Nian Lin, Franck Vidal, Aitor Landa, Mario Ruben, Johannes V. Barth, and Klaus Kern. Programming supramolecular assembly and chirality in two-dimensional dicarboxylate networks on a cu(100) surface. *Nano Letters*, 5(5):901–904, may 2005. doi: 10.1021/nl050362a. URL https://doi.org/10.1021%2Fnl050362a.

[85] Baron William Thomson Kelvin. *The Molecular Tactics of a Crystal*. Clarendon Press, 1894. ISBN 1363972537. URL http://www.ebook.de/de/product/27145541/molecular_tactics_of_a_crystal.html.

[86] Patrick J. Morandi. The classification of wallpaper patterns: From group cohomology to escher's tessellations, 2003. URL http://sierra.nmsu.edu/morandi/notes/Wallpaper.pdf.

[87] P Engel, L Michel, and M Senechal. Lattice Geometry. Technical Report IHES-P-2004-45, Inst. Hautes Etud. Sci., Bures-sur-Yvette, 2004. URL http://cds.cern.ch/record/859509.

[88] Donald E Knuth. Dancing links. *Millenial Perspectives in Computer Science*, 18(arXiv:cs/0011047. KNUTH MIGRATION 11-2004):4, Sep 2009. URL http://cds.cern.ch/record/1208945.

[89] Richard M. Karp. Reducibility among combinatorial problems. In *Complexity of Computer Computations*, pages 85–103. Springer US, 1972. doi: 10.1007/978-1-4684-2001-2_9. URL https://doi.org/10.1007/978-1-4684-2001-2_9.

[90] Yanxi Liu and Robert Collins. Frieze and wallpaper symmetry groups classification under affine and perspective distortion. Technical Report CMU-RI-TR-98-37, Pittsburgh, PA, July 1998. URL http://ri.cmu.edu/publications/frieze-and-wallpaper-symmetry-groups-classification-under-affine-and-perspective-distortion/.

[91] Peter Gruber. *Handbook of Convex Geometry*. Elsevier Science, Amsterdam New York, 1993. ISBN 9780080934402. URL http://www.ebook.de/de/product/23510461/author_unknown_handbook_of_convex_geometry.html.

[92] Rainer Storn and Kenneth Price. Differential evolution - a simple and efficient heuristic for global optimization over continuous spaces. *Journal of Global Optimization*, 11(4):341–359, 1997. doi: 10.1023/A:1008202821328. URL https://doi.org/10.1023%2Fa%3A1008202821328.

[93] Daan Wierstra, Tom Schaul, Tobias Glasmachers, Yi Sun, Jan Peters, and Jürgen Schmidhuber. Natural evolution strategies. *Journal of Machine Learning Research*, 15:949–980, 2014. URL http://jmlr.org/papers/v15/wierstra14a.html.

[94] Thomas A. Runkler. *Brief Review of Some Optimization Methods*, pages 123–126. Vieweg+Teubner Verlag, Wiesbaden, 2012. ISBN 978-3-8348-2589-6. doi: 10.1007/978-3-8348-2589-6_10. URL http://dx.doi.org/10.1007/978-3-8348-2589-6_10.

[95] Melanie Mitchell. *An Introduction to Genetic Algorithms*. MIT Press, Cambridge, MA, USA, 1998. ISBN 0262631857.

[96] Mohammad Shafiul, Raiyan Yousuf, Faria Alam, and Hossain Shaikh. Experimental comparison between differential evolution and artificial bee colony algorithm: A case study with continuous optimization problems. *International Journal of Applied Information Systems*, 10(4):35–39, jan 2016. doi: 10.5120/ijais2016451493. URL https://doi.org/10.5120%2Fijais2016451493.

[97] Sorin Istrail, Alan Hurd, Ross A. Lippert, Brian Walenz, Serafim Batzoglou, John H. Conway, and Fredrick W. Peyerl. Prediction of self-assembly of energetic tiles and dominos: Experiments, mathematics and software. Technical report, 2000. URL http://www.brown.edu/Research/Istrail_Lab/papers/10.1.1.134.9581.pdf.

[98] C. Bradford Barber, David P. Dobkin, and Hannu Huhdanpaa. The quickhull algorithm for convex hulls. *ACM Transactions on Mathematical Software*, 22 (4):469–483, dec 1996. doi: 10.1145/235815.235821. URL https://doi.org/10.1145/235815.235821.

[99] Dmytro Antypov and James A. Elliott. Wang-landau simulation of polymer-nanoparticle mixtures. *Macromolecules*, 41(19):7243–7250, oct 2008. doi: 10.1021/ma8012652. URL https://doi.org/10.1021/ma8012652.

[100] F. P. Kelly. *Reversibility and Stochastic Networks*. CAMBRIDGE UNIV PR, 2011. ISBN 1107401151. URL http://www.ebook.de/de/product/14639260/f_p_kelly_reversibility_and_stochastic_networks.html.